

A High-Order Low-Mach Number AMR Construction for Chemically Reacting Flows

Cosmin Safta, Jaideep Ray, Habib N. Najm

Sandia National Laboratories, Livermore, CA

Abstract

A high-order projection scheme was developed for the study of chemically reacting flows in the low-Mach number limit. The numerical approach for the momentum transport uses a combination of cell-centered/cell-averaged discretizations to achieve a fourth-order formulation for the pressure projection algorithm. This scheme is coupled with a second-order in time operator-split stiff approach for the species and energy equations. The code employs a fourth-order, block-structured, adaptive mesh refinement approach to address the challenges posed by the large spectrum of spatial scales encountered in reacting flow computations. Results for advection-diffusion-reaction configurations are used to illustrate the performance of the numerical construction.

Keywords:

1. Introduction

Since the detailed structure and various non-equilibrium characteristics of chemically reacting systems are difficult and costly to obtain experimentally, numerical simulation is an important tool in complementing experimental investigations of combustion processes. Chemical reacting systems based on hydrocarbon fuels typically exhibit a large spectrum of characteristic spatial and temporal scales. The complexity of kinetic models even for simple hydrocarbon fuels compounds this problem making multidimensional numerical simulations difficult even for laboratory scale configurations.

These difficulties are commonly addressed in a variety of ways. For low speed flows, one may adopt a low Mach number approximation of the Navier-Stokes equations [1] for the momentum transport. This approximation assumes that acoustic waves travel at infinite speed, a justifiable assumption in many low-speed flows. One can also exploit the structure of the governing equations and adopt an operator-split construction, performing the transport and reactive time-advancement

via specialized integrators [2]. In problems where fine structures exist only in a small fraction of the domain e.g. in laminar jet flames, one may employ adaptive mesh refinement (AMR) [3] to concentrate resolution only where needed [4, 5, 6, 7], while maintaining a coarse mesh resolution elsewhere.

We have recently developed a numerical model that aims to address some of the challenges posed by the use of AMR for reacting flow computations. In order to reduce the number of grid points and the number of refinement levels in the computational mesh hierarchy, we employ high-order stencils to discretize the transport equations and to interpolate between the computational blocks on adjacent mesh levels. Further, we employ a projection scheme for the momentum transport on a *uniform* mesh, that is coupled with the adaptive mesh solution of the scalar transport equations. This hybrid construction is driven by a number of practical considerations, as further detailed in the next section. For ease of implementation with AMR, we employ an extended-stability Runge Kutta Chebyshev (RKC) scheme [8] to time-advance the system; the timestep size is constrained by the Fourier number (“diffusion CFL”). The 2D numerical scheme developed in this investigation is designed to work with block-structured adaptively refined meshes (alternatively, structured adaptive mesh refinement, SAMR). The 3D implementation of the numerical construction is underway and results will be presented in a subsequent paper.

The paper is structured as follows. In Section 1.1 we describe SAMR and show why its details prevent a straightforward adoption of high-order spatial discretizations. We also describe the basic structure of the algorithm pursued here, in terms of the mesh structures used for different variables. In Section 2, we review current research on the development of high-order methods for SAMR computations. This also includes a review of literature on the pairing of interpolation and derivative stencils. In Section 3 we develop the numerical scheme, including a novel construction for solving the pressure equation, which follows an existing finite volume construction. In Section 4, we present results. We demonstrate that we preserve second-order accuracy in time and achieve fourth-order in space on mesh hierarchies. Numerical experiments are conducted on canonical advection-diffusion configurations, 1D premixed flames and a 2D interaction of a vortex pair with a planar flame. Conclusions are in Section 5.

1.1. SAMR and the Statement of the Problem

A detailed description of SAMR can be found in [3]; we provide a summary here. SAMR meshes are generally employed in discretizing rectangular domains. A relatively coarse Cartesian mesh is laid and various field variables (e.g., temperature, species mass fractions) are initialized on it. The mesh is generally too coarse to properly resolve all structures and based on an error metric (often a threshold on a spatial derivative), grid cells requiring further refinement are marked. These grid cells are collated into rectangular patches and a finer grid (often twice as dense as the initial grid, obtained by splitting each cell in half in each dimension) is described on it. Since regions requiring refinement can be disjoint, a multitude of patches may be formed; they are referred to as *children* patches. This refinement procedure is performed recursively, until a hierarchy of patches is formed. The bottom level of the hierarchy (often called the zeroth level) is the original coarse mesh. Note that children patches are not *embedded* in the parent mesh but rather *overlayed* on it; thus the grid hierarchy is a set of rectangular Cartesian meshes. Also note that children patches obtained in this manner may abut but do not intersect their siblings.

The rectangular nature of the patches and the Cartesian mesh allows the evaluation of high order stencils in a straightforward manner. A layer of “ghost” cells are added to each patch on each refinement level to allow the use of symmetric stencils. Ghost cells are initialized by copying the corresponding values from adjacent patches on the same refinement level. Ghost cells, that do not overlap patches with the same refinement, are initialized by interpolating the values from immediately coarser levels. Note that coarse-fine interpolations have to be performed with care; they have the potential to destroy the order of accuracy, especially since the interpolated values are directly (and only) used to calculate derivatives.

The adaptive nature of SAMR arises from the fact that the process of identifying cells requiring refinement (followed by the addition of finer patches) can be performed periodically, allowing the simulation to track steep gradients, while simultaneously coarsening regions which no longer require fine patches. A recursive “time-refinement” [3, 9] is used to advance the solution on the mesh hierarchy.

The development of a finite-difference numerical scheme that combines RKC, fourth-order spatial accuracy and a projection scheme in a SAMR context has required significant algorithmic development and has progressed in stages. In [9], we coupled RKC with SAMR and fourth-order

spatial discretizations to solve a reaction-diffusion system on SAMR meshes. Using a non-stiff H_2 – air chemical mechanism we empirically demonstrated that the desired spatial and temporal accuracies could be achieved. In [10], we investigated how high-order interpolation and derivative stencils ought to be paired in order to achieve a desired order of convergence. In this paper, we bring together these results to construct a numerical scheme that includes a projection method to solve the low Mach approximation of the Navier-Stokes equations within the context of reacting flows. We also demonstrate the method with a methane-air chemical mechanism.

In this paper, we address the challenge of coupling the high-order approach for solving reaction-diffusion systems in [9, 10] to a projection method for solving the low Mach number form of the Navier-Stokes equations. We implement a construction in which the momentum equations are discretized and solved on a uniform mesh only, and are coupled to the solution of the species conservation and energy equations on an adaptive mesh hierarchy. This construction is driven by a number of practical considerations. A key factor is that we are primarily concerned with detailed computations of reacting flows with complex (and stiff) chemical kinetics; such that we typically handle hundreds of species. **As a result, the momentum solution for the velocity and pressure fields is typically responsible for a small fraction of the overall computational cost per mesh cell.** Further, since, in many low speed reacting flows of interest, the smallest relevant spatial length scales of the velocity and pressure fields are significantly larger than those of the scalar fields in the primary reaction zone, and it is the accuracy in resolving the *scalar* length scales, that ultimately determines the requisite finest mesh-resolution in an AMR construction. Accordingly, the mesh resolution required for accurate solution of the momentum equations is much coarser than that necessary for comparable accuracy in the solution of the scalars. The relatively coarse mesh resolution requirement, and the low computational cost per mesh cell, means that solving the momentum equations on a uniform mesh is not expected to significantly increase overall computational cost. Further, this construction is, in fact, expected to have superior efficiency, since the elliptic solver required for the pressure equation is more efficient on a uniform mesh compared to a multilevel one [7, 11, 12]. This construction retains the overall low computational cost of the momentum solution relative to the cost of time integration of the scalar fields. It is designed to be optimal for reacting flows, targeting momentum/scalar elements of the solution with suitably chosen mesh structures. It is a *multihierarchy* construction where the momentum equations are handled on a

uniform mesh –of adequate resolution– and are coupled to the adaptive mesh solution of the species and energy equations. The uniform mesh is chosen to be sufficiently resolved for the velocity and pressure fields. Note that such constructions have been used in other contexts where necessary, for example the methods developed in numerical relativity for simulating relativistic, self-gravitating fluids [13, 14].

In summary, the mesh on which the momentum equations are solved and the lowest level of the mesh hierarchy resolving the scalar equations are identical. The numerical methods used to discretize them are also the same. The time-stepping scheme for the momentum and scalar equations employs an operator-split construction. The velocities are interpolated up from the lowest SAMR level, when required for convecting the scalars on finer SAMR levels, using interpolations that have been previously verified [10]. The main contributions of the paper, then, are:

1. The construction of a projection scheme, especially a formulation for the pressure that is consistent, stable and preserves fourth-order accuracy. This is demonstrated empirically over a set of 1D and 2D test cases.
2. The implementation of this projection scheme in a coupled high order low Mach number reacting flow construction, with a SAMR discretization for species mass fractions and temperature fields, that preserves overall fourth-order accuracy. Convergence tests, using errors calculated vis-à-vis both analytical and spatially refined solutions are employed for this purpose.

We note that this construction does not employ interpolation stencils to explicitly ensure conservation of fluxes at coarse-fine boundaries (see [15] for the stencils for a fourth-order Poisson equation solver on SAMR meshes). While conservation is a desired property, its manifestation in resolved high-order computations is primarily as a component of the truncation error budget. A fully resolved velocity field on the coarsest mesh level, coupled with sixth-order interpolations at coarse-fine boundaries should be sufficient for preserving conservation of fluxes *up to discretization order*. We check this empirically by performing an advective-diffusive test of a scalar “blob”, making it travel across a number of coarse-fine interfaces. The numerical solution is compared against an analytical one to assess the effect of approximate preservation of flux continuities. This may be considered to be a reacting-flow analog of the acoustic pulse test performed in [16] for much the same purpose. Later, we re-examine the same problem in a more stringent context, by subjecting

a premixed flame to a large, vortically-induced stretch and investigating its effect on the spatial distribution of radicals.

2. Literature Review

SAMR was pioneered by Berger and Colella in their 1989 paper on the simulation of shock waves [3]. Thereafter, it has been used in the simulation of flames with complex chemistry [17] in a variety of realistic 3D laboratory configurations [18, 19]. The numerical schemes employed in these investigations were of second-order accuracy, except in the case of shock-wave simulations, where the use of limiters reduce the accuracy to first-order in the vicinity of the discontinuities. In fact, literature on the construction of high-order (i.e., greater than second-order) accuracy on SAMR grids is very sparse.

Barad and Colella [15] addressed the problem of solving the Poisson equation on 2D and 3D SAMR meshes. Starting with a classical Mehrstellen method, they developed a fourth-order construction. Tests were done on a 2-level mesh (i.e., a coarse mesh with a single level of refinement) and fourth-order convergence of the numerical discretization error was demonstrated. The main contribution of this method lies in the treatment of the interpolations at the interface between coarse and fine level patches. A sequence of one-dimensional interpolations, some fourth-order and others sixth-order, were required to obtain the desired order properties. The sequence of interpolations is dependent on the configuration of patches. The refinement ratio between parent and children patches is required to be four. Recently, Colella and coworkers have investigated developing fourth-order SAMR methods for flows with shock waves. In such flows, the use of limiters at the discontinuity usually renders the scheme first-order, and consequently schemes employing them were designed to preserve only second-order accuracy in the smooth portions of the flow. In [20], Colella and Sekora developed a limiter that preserves accuracy at smooth extrema, allowing a rationale for developing higher-order finite-volume schemes for SAMR. Preliminary results on SAMR meshes are in [21]. The determination of data in the “halo” surrounding fine patches is done by solving a constrained least-squares problem (the constraints are the conservation laws). Results are presented for both Cartesian and mapped SAMR grids.

Huang et al. [16, 22] used fourth-order dispersion-relation preserving schemes on SAMR meshes to solve a coupled set of three linear equations to investigate acoustic radiation from an aero-engine

intake. In their case, the “halo” of grid cells around a fine patch was filled (using data from the underlying coarse parent patch) using symmetric fourth-order interpolations. Tests showed that a Gaussian pulse propagating across a coarse-fine interface showed far smaller oscillations when fourth-order interpolants were used. The oscillations could be removed using either filters or artificial viscosity, and separate tests were done to demonstrate this. Preference for artificial viscosity was indicated since it can be user-controlled. Tests were limited to 2D and three-level SAMR meshes were used. The construction was tested on both Cartesian SAMR meshes as well as after mapping the grid to be body-fitted i.e., to be fitted to the engine inlet.

A *multihierarchy* construction was employed in the simulation of relativistic, gravity-driven flows [13, 14]. These interactions are modeled by coupling Einstein’s (elliptic) field equations with relativistic hyperbolic Euler equations for inviscid flow. The fields evolved using Einstein’s equations are smooth and show large structures only. The gravitational/tidal effects on matter lead to relativistic fluid dynamical phenomena, including shocks. In [13] the authors couple a spectral method for Einstein’s fields with a Godunov solver for the Euler equations. Second-order accuracy was empirically observed, with adequate filtering to remove oscillations associated with polynomial interpolations between the two solvers. In [14] a pseudospectral formulation was adopted, primarily to simplify the interpolation between the field and hydrodynamical meshes, and second-order convergence was also observed. The interpolation order was generally third-order. In both cases, the size of the hydrodynamical and field grids (in each direction) was within a factor of 2 of each other; however, given the difference in their extent (factor of 20) their spatial resolutions were vastly different.

This paper builds on our previous efforts to develop tools for high-order SAMR computations. In [9] we adapted RKC for use on SAMR meshes and verified that its order properties were preserved by testing it with a reaction-diffusion system; in [23] we investigated its behavior when advection was added to the system. In [10] we derived expressions for fourth (and higher) order finite-difference stencils, both symmetric and skewed, for evaluating derivatives as well as for interpolations. We then investigated how derivatives and interpolation stencils needed to be paired in order to obtain a desired order of convergence in the spatial discretization error. This was performed empirically, using a set of model equations. The models were chosen so that the maximum spatial derivatives in the equation were different. Tests showed that the pairing required to obtain an accuracy commensurate

with the derivative stencil required an interpolation stencil whose order was higher than that of the derivative stencil; further, this excess was related to the highest spatial derivative in the equation. An expression relating the three was obtained. Tests also revealed that the use of high-order methods with SAMR could lead to Gibbs phenomena at the interface between coarse and fine patches, but which could be removed by filtering. The correct pairing of derivative stencil and filter orders (which would not degrade the order properties of the error convergence under mesh refinement) was obtained via a expression relating the two. All tests were performed with nodal variables.

With this introduction and extensive background, we proceed in the following to the description of the numerical construction.

3. Numerical Methodology

In the low-Mach number limit, the continuity, momentum and scalar equations are written in compact form as

$$\nabla \cdot \mathbf{v} = -\frac{1}{\rho} \frac{D\rho}{Dt} \quad (1a)$$

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{1}{\rho} \nabla p + C_U + D_U \quad (1b)$$

$$\frac{\partial T}{\partial t} = C_T + D_T + S_T \quad (1c)$$

$$\frac{\partial Y_k}{\partial t} = C_{Y_k} + D_{Y_k} + S_{Y_k} \quad k = 1, 2, \dots, N_s \quad (1d)$$

Here \mathbf{v} is the velocity vector, ρ the density, T the temperature, Y_k the mass fraction of species k , p is the hydrodynamic pressure, and N_s is the number of chemical specie. The $\frac{D}{Dt}$ operator in the continuity equation represents the material derivative, $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla$. The system of governing equations is closed with the equation of state for an ideal gas:

$$P_0 = \frac{\rho \mathfrak{R} T}{\overline{W}} = \rho \mathfrak{R} T \sum_{k=1}^{N_s} \frac{Y_k}{W_k} = \text{const} \quad (2)$$

where P_0 is the thermodynamic pressure, \mathfrak{R} is the universal gas constant, W_k is the molecular weight of species k , and \overline{W} is the molecular weight of the mixture. The thermodynamic pressure is spatially uniform in the low-Mach number limit. Further, restricting our focus to flows in open

domains, P_0 is constant.

The convection and diffusion terms in (1) are given by

$$C_U = -(\mathbf{v} \cdot \nabla) \mathbf{v}, \quad D_U = \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau}, \quad (3a)$$

$$C_T = -(\mathbf{v} \cdot \nabla) T, \quad D_T = \frac{1}{\rho c_p} \nabla \cdot (\lambda \nabla T) - \frac{1}{c_p} \left(\sum_{k=1}^{N_s} c_{p,k} Y_k \mathbf{V}_k \right) \cdot \nabla T \quad (3b)$$

$$C_{Y_k} = -(\mathbf{v} \cdot \nabla) Y_k, \quad D_{Y_k} = -\frac{1}{\rho} \nabla \cdot (\rho Y_k \mathbf{V}_k) \quad (3c)$$

and the source terms by

$$S_T = -\frac{1}{\rho c_p} \sum_{k=1}^{N_s} h_k \dot{\omega}_k, \quad S_{Y_k} = \frac{\dot{\omega}_k}{\rho} \quad (4)$$

Here, $\boldsymbol{\tau}$ is the stress tensor given by $\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \nabla \cdot \mathbf{v} \right)$, μ is the dynamic viscosity, and λ is the mixture thermal conductivity. Further,

$$\mathbf{V}_k = -\frac{D_{k,m}}{Y_k} \left(\nabla Y_k + \frac{Y_k}{\bar{W}} \nabla \bar{W} \right),$$

is the diffusion velocity of species k , where $D_{k,m}$ is the mixture-averaged diffusivity of species k . Finally, c_p and $c_{p,k}$ are the specific heats at constant pressure for the mixture and species k , respectively, and h_k and $\dot{\omega}_k$ are the specific enthalpy and molar production rate, respectively, of species k .

NASA polynomials [24] are used to compute thermodynamic properties. The mixture-averaged transport properties are evaluated using a dipole-reduced formalism [25]. The Soret and Dufour effects are negligible for hydrocarbon combustion and are not included in the transport model.

In the numerical construction presented below, the equation of state (2) is used to derive an expression for the right hand side of the continuity equation (1a)

$$\frac{DP_0}{Dt} = 0 \rightarrow \frac{1}{\rho} \frac{D\rho}{Dt} = -\frac{1}{T} \frac{DT}{Dt} - \sum_{k=1}^{N_s} \frac{\bar{W}}{W_k} \frac{DY_k}{Dt} \quad (5)$$

$$= -\frac{1}{T} (D_T + S_T) - \sum_{k=1}^{N_s} \frac{\bar{W}}{W_k} (D_{Y_k} + S_{Y_k}) \quad (6)$$

3.1. Numerical Approach

In this section we develop a numerical method to solve the low Mach number reacting flow equations described above. The divergence constraint (eq. 1a) resulting from the low Mach number limit leads to a differential algebraic equation system, for which we adopt a projection method to solve for the velocity and pressure fields. The momentum solver is coupled with a solver for the species and temperature fields, arriving at an overall construction that is fourth-order in space, second-order in time.

As described in Section 1, the problem is solved on a mesh hierarchy. On a given patch, variables are defined at cell centers and edge-centers. A detail of a 2D computational grid is shown in Fig. 1. The computational domain is composed of “cells” and “edges”. The temperature, density, pressure, and species mass fractions are located at cell centers, shown with filled circles, while the velocity components are located at edge centers. The cell-centers correspond to integer indices, i.e. (i, j) , $(i + 1, j)$, while the edges correspond to fractional indices, i.e. $(i - \frac{1}{2}, j)$, $(i + \frac{1}{2}, j)$ for the x -velocity and $(i, j - \frac{1}{2})$, $(i, j + \frac{1}{2})$ for y -velocity. Figure 2 shows the cell and edge correspondence between overlapping grids on two consecutive mesh levels.

The numerical integration of the system of equations is performed in three stages. First, a projection approach is adopted to advance the velocity field based on the equations (1a-1b). The projection scheme is implemented on a uniform mesh. In the second stage, the scalars are advanced using an operator split approach that separates the convection and diffusion contributions from the ones due to the chemical source terms. We implement a symmetric Strang splitting scheme beginning with the source term contribution for half the time step, followed by the contributions from convection and diffusion terms for a full time step, and concluded by the remaining contribution from the reaction term for half the time step. During this stage, scalars are recursively advanced on successively refined grids necessary to resolve the scalar spatial structures. The time stepping is concluded with the third stage, which repeats the projection algorithm on a uniform mesh using information on scalar fields obtained at the end of the second step. The algorithm is described in detail below.

Stage 1_a

The 2nd order Adams-Bashforth scheme is used to advance the velocity field using momentum and

diffusion terms only

$$\frac{\hat{\mathbf{v}}^{n+1} - \mathbf{v}^n}{\Delta t} = \left(1 + \frac{1}{2} \frac{\Delta t}{\Delta t_o}\right) (C_U^n + D_U^n) - \frac{1}{2} \frac{\Delta t}{\Delta t_o} (C_U^{n-1} + D_U^{n-1}) \quad (7)$$

Superscripts n and $n - 1$ refer to values at the current t^n and previous t^{n-1} times, respectively. The above expression takes into account changes in time step values, $\Delta t = t^{n+1} - t^n$ and $\Delta t_o = t^n - t^{n-1}$.

The convection terms contain components that are either collocated or staggered. For the x -velocity, $C_U|_x = -u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y}$, the first term is a product of collocated variables since a centered stencil is used to obtain $\frac{\partial u}{\partial x}$ at $(i \pm \frac{1}{2}, j)$. For the second part, v is interpolated from edge centers $(i, j \pm \frac{1}{2})$ to edge centers $(i \pm \frac{1}{2}, j)$ corresponding to u -velocity and $\frac{\partial u}{\partial y}$ is computed using collocated stencils. Velocity derivatives are discretized using 4th order stencils. A 6th order accurate interpolation stencil is used in order to preserve the overall 4th order accuracy of the scheme. The convection terms for the other velocity components are computed in a similar fashion. The stencils for these discretizations are given elsewhere [10].

The evaluation of the viscous stress tensor components involves staggered stencils for velocity derivatives and interpolants for the mixture viscosity. The diagonal terms τ_{ii} are located at cell centers, while the off-diagonal ones $\tau_{ij, i \neq j}$ are located at cell vertices, shown with open symbols in Fig. 1. The mixture viscosity is first calculated based on species mass fractions and temperature at cell-centers using a mixture-averaged formulation [25]. The viscosity needed for the off-diagonal stress tensor terms is computed by interpolation from the cell center values, using a 6th order stencil. The density at edge centers necessary to compute D_U is also computed by interpolation from cell center values. The stencils necessary for these interpolations can be found in [10], while the 4th order staggered stencils for velocity derivatives are given in Appendix B.

Stage 1_b

The provisional velocity field, $\hat{\mathbf{v}}$, does not satisfy eq. (1a). This equation is used in conjunction with eq. (1b) to derive an equation for the hydrodynamic pressure field

$$\nabla \cdot \left(\frac{1}{\rho^{n+1}} \nabla p \right) = \frac{1}{\Delta t} \left(\nabla \cdot \hat{\mathbf{v}}^{n+1} + \frac{1}{\rho} \frac{D\rho}{Dt} \Big|^{n+1} \right), \quad (8)$$

that will be then used to correct the provisional velocity field. Since the scalar fields at t^{n+1} are not

yet known, $\left. \frac{1}{\rho} \frac{D\rho}{Dt} \right|^{n+1}$ is estimated by extrapolation:

$$\left. \frac{1}{\rho} \frac{D\rho}{Dt} \right|^{n+1} = \left(1 + \frac{\Delta t}{\Delta t_o} \right) \left. \frac{1}{\rho} \frac{D\rho}{Dt} \right|^n - \frac{\Delta t}{\Delta t_o} \left. \frac{1}{\rho} \frac{D\rho}{Dt} \right|^{n-1} \quad (9)$$

The numerical evaluation of $\frac{1}{\rho} \frac{D\rho}{Dt}$ is described in Stage 3_b. The density at t^{n+1} , ρ^{n+1} , is also extrapolated from values at t^n and t^{n-1} similar to eq. (9). Finally, the fourth-order discretization of the pressure equation (8) is described in Section (3.2).

Stage 1_c

The gradient of the hydrodynamic pressure is used to correct the provisional velocity field $\hat{\mathbf{v}}^{n+1}$ to obtain the predicted velocity at $n + 1$

$$\mathbf{v}^{n+1,p} = \hat{\mathbf{v}}^{n+1} - \frac{\Delta t}{\rho^{n+1}} \nabla p, \quad (10)$$

On the boundaries of the computational domain, the pressure gradient normal to the boundary is set to zero in **Stage 1_b** which results in $\mathbf{v}^{n+1,p}|_n = \hat{\mathbf{v}}^{n+1}|_n$. Here subscript n denotes the velocity component normal to the boundary. Superscript p was added to \mathbf{v} to distinguish the predicted velocity values obtained at the end of Stage 1 from the corrected ones obtained at the end of Stage 3 below. The boundary conditions for the velocity and hydrodynamic pressure are detailed in Section 3.4.

Stage 2_a

In the first part of the second stage, temperature and species mass fractions are advanced over half the time step based on contributions from the source terms, S_T and S_{Y_k} .

$$\begin{aligned} T^* - T^n &= \int_{\Delta t/2} S_T dt \\ Y_k^* - Y_k^n &= \int_{\Delta t/2} S_{Y_k} dt \quad k = 1, 2, \dots, N_s \end{aligned} \quad (11)$$

The CVODE stiff integrator package [26] is used to integrate eqs. (11). Since during this stage the *rhs* in eq. (11) has no spatial dependence, the time advancement is done independently for each computational grid cell. At the end of the stage, the scalar values are recursively restricted from fine to coarse grid levels. Stencils for interpolations between coarse and fine grid levels are given in

Appendix B.

Stage 2_b

During the second part of the second stage, a 2nd-order, multi-stage, Runge-Kutta-Chebyshev (RKC) [8] scheme is used to advance scalars based on the contributions from convection and diffusion terms:

$$\begin{aligned} T^{**} - T^* &= \int_{t^n}^{t^{n+1}} \underbrace{C_T + D_T}_{F_T} dt \\ Y_k^{**} - Y_k^* &= \int_{t^n}^{t^{n+1}} \underbrace{C_{Y_k} + D_{Y_k}}_{F_{Y_k}} dt \quad k = 1, 2, \dots, N_s \end{aligned} \quad (12)$$

The numerical details for the multi-stage RKC scheme are given elsewhere [2, 8].

RKC is developed to accomodate large diffusive CFL number stability, which makes it suitable for diffusion-dominated configurations which are of interest for this project. For these configurations the imaginary parts of the eigenvalues due to the convection are small while the negative real eigenvalues due to diffusion are much larger in magnitude. Alternate time discretization schemes that accomodate larger advection terms can be employed, *e.g.* utilizing a second level of Strang splitting for the convection terms and using a suitable associated RK scheme.

The stability limit for the time step Δt depends quadratically on the number of RKC stages. A detailed analysis of the range of values for the time step and number of RKC stages, typically between 2 and 32, is provided in [2].

As scalars are advanced from t^n to t^{n+1} , velocity values needed to construct advection fluxes for the intermediate RKC times are computed by interpolation based on the values at t^n and $\mathbf{v}^{n+1,p}$ at $t^{n+1} = t^n + \Delta t$:

$$\mathbf{v}^{(s)} = (1 - c_s) \mathbf{v}^n + c_s \mathbf{v}^{n+1,p}.$$

Here, $c_s = (t_s - t_n) / \Delta t$ is the time fraction corresponding to RKC stage s , $1 \leq s \leq M$. As mentioned above, the number of RKC stages M controls the stability regions of the scheme. Similar to some of the velocity convection terms, scalar convection terms involve components that are not collocated. For these terms the scalar derivatives are evaluated at the same location as the velocity components using 4th order staggered derivative stencils. The products $u_i \frac{\partial \Phi_i}{\partial x_i}$ are then interpolated from edge

centers to cell centers. The interpolation stencils for these discretizations are derived in [10], while the staggered derivatives are provided in Appendix B.

The diffusion terms D_T and D_{Y_k} are computed using high-order derivatives and interpolations as follows:

- Temperature and species mass fractions are interpolated from cell centers to edge centers, and transport properties, e.g. diffusion coefficients, are evaluated at edge centers based on these interpolated values.
- The temperature conduction term is evaluated using staggered derivative stencils for T and then again for $\lambda \nabla T$.
- The diffusion velocity is computed at edge centers using staggered derivatives stencils.
- The components of the dot product in the second part of D_T are computed at edge centers and then interpolated to cell centers.
- The species diffusion term is computed similar to the thermal conduction term in the temperature equation.

High-order interpolation stencils can be found in [10] while staggered derivative stencils are provided in Appendix B. The above algorithm results in the computation of convective and diffusive terms at cell centers. They are then used to advance the scalar values with the RKC algorithm.

Time advancement of the mesh hierarchy. SAMR is used to adaptively refine the computational grid in regions where the combustion is active and the internal flame structure needs to be resolved accurately. Figure 3 shows a schematic of the time advancement procedure, first proposed by Berger and Collela [3], on adjacent mesh levels, L and $L + 1$, in the grid. The scalars are first advanced on the coarse level L using the RKC algorithm. After the advancement is completed on L , the solution on this level is used to provide boundary conditions (via coarse-to-fine prolongation) for the solution advancement on $L + 1$.

The values at various intermediate times between t^n and $t^n + \Delta t$ are computed by interpolation on level L and the results are interpolated to level $L + 1$. Ray *et al.* [10] showed that, in order to preserve the overall order of accuracy, the interpolation order needs to account for the highest spatial

derivative in the discretization. For our numerical construction this condition leads to 6th order stencils for interpolating data between adjacent grid levels.

The grid size on the finer mesh level $L + 1$ is half compared to L and for stability purposes the time step is halved. At the end of the two sub-steps on $L + 1$ the fine-grid solution is presumably “better” and is interpolated to the coarse grid L . Boundary conditions for the scalar advance are provided in Section 3.4.

Stage 2_c

Stage 2_c is a repeat of Stage 2_a, using the “**” scalar values as initial conditions

$$\begin{aligned} T^{n+1} - T^{**} &= \int_{\Delta t/2} S_T dt \\ Y_k^{n+1} - Y_k^{**} &= \int_{\Delta t/2} S_{Y_k} dt \quad k = 1, 2, \dots, N_s \end{aligned} \quad (13)$$

At the end of this stage all scalars correspond to t^{n+1} . The solution advancement during stages 2_a and 2_c is based on the source terms which are independent of the solution at neighboring grid points. Consequently no prolongations are required for these stages. However, after each of these stages are completed, the solution needs to be restricted (fine-to-coarse grid interpolation) recursively starting from the finest grid level in the main hierarchy.

Stage 3_a

The provisional velocity field values at t^{n+1} are re-evaluated based on the scalar values obtained at the end of Stage 2 and on the predicted velocity values at the end of Stage 1

$$\frac{\hat{\mathbf{v}}^{n+1} - \mathbf{v}^n}{\Delta t} = \frac{1}{2} ((C_U^n + D_U^n) + (C_U^{n+1} + D_U^{n+1})) \quad (14)$$

The convection C_U^{n+1} and diffusion D_U^{n+1} are based on the velocity field $\mathbf{v}^{n+1,p}$ and the scalar values at t^{n+1} .

Stage 3_b

The hydrodynamic pressure field is re-computed using equation (8). The divergence term that enters the rhs for this equation is constructed using the provisional velocity field obtained in Stage 3_a, while eq. (6) is used to compute $\left. \frac{1}{\rho} \frac{D\rho}{Dt} \right|^{n+1}$ based on scalar values at t^{n+1} , obtained in Stage 2. The discretization of the diffusion terms in eq. (6) is similar to the discretization of the diffusion

terms described above in Stage 2_b.

Stage 3_c

This stage is similar to Stage 1_c. The gradient of the hydrodynamic pressure obtained at Stage 3_b is used to correct $\hat{\mathbf{v}}^{n+1}$ (computed at Stage 3_a) to obtain \mathbf{v}^{n+1} .

3.2. Discretization of the Pressure Equation

We present in Appendix A a fourth-order finite difference discretization for the variable coefficient Poisson equation (8). We show that additional manipulations of the linear system for the pressure solve, and consequently the boundary conditions for the provisional velocity field, are required in order to obtain consistent linear systems.

In this section we present a formulation for the pressure solve based on cell- and edge-averaged values of flow variables. This formulation retains a high-order convergence rate while simplifying the consistency condition for the linear system. This methodology is similar to Kadioglu *et al* [27] and relies on the fact that the cell-averaged divergence of a vector field is identically equal to the difference between its edge-averaged values. A schematic of the locations of the cell- and edge-centered variables and their cell- and edge-averaged counterparts is shown in Fig. 4. In the expressions below, a horizontal bar is used for cell-averaged values while tilde is used for edge averages.

The cell average of the velocity divergence in the *rhs* of (8) is given by

$$\overline{\nabla \cdot \mathbf{v}}_{i,j} = \frac{\tilde{u}_{i+1/2,j} - \tilde{u}_{i-1/2,j}}{h_x} + \frac{\tilde{v}_{i,j+1/2} - \tilde{v}_{i,j-1/2}}{h_y} \quad (15)$$

Fourth-order Lagrange polynomial interpolants are fitted through the edge-centered velocities and then used to compute the edge-averaged velocity. For edge $(i + \frac{1}{2}, j)$, the averaged normal velocity is computed as

$$\tilde{u}_{i+\frac{1}{2},j} = \frac{-17u_{i+\frac{1}{2},j-2} + 308u_{i+\frac{1}{2},j-1} + 5178u_{i+\frac{1}{2},j} + 308u_{i+\frac{1}{2},j+1} - 17u_{i+\frac{1}{2},j+2}}{5760} \quad (16)$$

The expressions for the *v*-velocities are similar, except that summation takes place in the *i*-direction.

The cell-averaged value for $\frac{1}{\rho} \frac{D\rho}{Dt}$ in the *rhs* of eq. (8) is computed by applying the 4-th order expression (16) to the cell-centered values, successively in the *i* and *j*-directions, respectively.

For simplicity of notation $\frac{1}{\rho}$ is replaced by “ a ” in the left-hand-side (*lhs*) of the pressure equation. The cell-averaged divergence $\overline{\nabla \cdot (a\nabla p)}$ is discretized based on (yet unknown) edge-averaged $(\widetilde{a\nabla p})$:

$$\overline{\nabla \cdot (a\nabla p)}_{i,j} = \frac{\widetilde{ap_{x_{i+\frac{1}{2},j}} - ap_{x_{i-\frac{1}{2},j}}}}{h_x} + \frac{\widetilde{ap_{y_{i,j+\frac{1}{2}}} - ap_{y_{i,j-\frac{1}{2}}}}}{h_y} \quad (17)$$

where p_x and p_y are the x - and y -components of the pressure gradient. The edge-averaged products $(\widetilde{ap_x})$ and $(\widetilde{ap_y})$ are computed as follows. First the cell-averaged \bar{a} is computed from cell-centered values similar to the cell-averaged $\frac{1}{\rho} \frac{D\rho}{Dt}$. The edge-averaged values for the inverse-density and pressure gradients fields on edge $(i + \frac{1}{2}, j)$ are then computed as

$$\begin{aligned} \tilde{a}_{i+\frac{1}{2},j} &= \frac{-\bar{a}_{i-1,j} + 7\bar{a}_{i,j} + 7\bar{a}_{i+1,j} - \bar{a}_{i+2,j}}{12} \\ \tilde{p}_{x_{i+\frac{1}{2},j}} &= \frac{\bar{p}_{i-1,j} - 15\bar{p}_{i,j} + 15\bar{p}_{i+1,j} - \bar{p}_{i+2,j}}{12h_x} \end{aligned} \quad (18)$$

In general, the edge-averaged product $(\widetilde{ap_x})$ is not equal to the product of edge-averages \tilde{a} and \tilde{p}_x . To determine a high-order expression for this product in terms of these edge-averages, power series expressions are developed to match the \tilde{a} and \tilde{p}_x on edges $(i + \frac{1}{2}, j)$, $(i + \frac{1}{2}, j \pm 1)$, and $(i + \frac{1}{2}, j \pm 2)$. A fourth-order expansion is written as:

$$f(y) = \sum_{n=0}^4 c^{(n)} (y - y_j)^n + o(h_y^4) \quad (19)$$

where y_j is the y -coordinate of the center of edge $(i + \frac{1}{2}, j)$. The coefficients $c^{(0)}, c^{(1)}, \dots$ are determined by matching the averaged power series expansions over the cell edge centered around y_j with edge-averaged values of the field approximated by this expansion. The first two coefficients in the above expansion corresponding to a are given by

$$\begin{aligned} c_a^{(0)} &= \frac{9\tilde{a}_{i+\frac{1}{2},j-2} - 116\tilde{a}_{i+\frac{1}{2},j-1} + 2134\tilde{a}_{i+\frac{1}{2},j} - 116\tilde{a}_{i+\frac{1}{2},j+1} + 9\tilde{a}_{i+\frac{1}{2},j+2}}{1920} \\ c_a^{(1)} &= \frac{5\tilde{a}_{i+\frac{1}{2},j-2} - 34\tilde{a}_{i+\frac{1}{2},j-1} + 34\tilde{a}_{i+\frac{1}{2},j+1} - 5\tilde{a}_{i+\frac{1}{2},j+2}}{48h_y} \end{aligned} \quad (20)$$

The average product $\widetilde{(ap_x)}$ over cell edge $(i + \frac{1}{2}, j)$ is then computed as

$$\widetilde{(ap_x)}_{i+\frac{1}{2},j} = \frac{1}{h_y} \int_{y_{j-1/2}}^{y_{j+1/2}} f_a f_{p_x} dy = c_a^{(0)} c_{p_x}^{(0)} + (c_a^{(0)} c_{p_x}^{(2)} + c_a^{(2)} c_{p_x}^{(0)} + c_a^{(1)} c_{p_x}^{(1)}) \frac{h_y^2}{12} + o(h_y^4) \quad (21)$$

where subscripts a and p_x were introduced to distinguish between the power series coefficients for a and ap_x , respectively. Making use of the power series above, the product of edge-averages $\tilde{a}_{i+\frac{1}{2},j} \cdot \tilde{p}_{x i+\frac{1}{2},j}$ can be written as

$$\tilde{a}_{i+\frac{1}{2},j} \cdot \tilde{p}_{x i+\frac{1}{2},j} = c_a^{(0)} c_{p_x}^{(0)} + (c_a^{(0)} c_{p_x}^{(2)} + c_a^{(2)} c_{p_x}^{(0)}) \frac{h_y^2}{12} + o(h_y^4)$$

This expression is substituted into the result for the average product (21) leading to

$$\widetilde{(ap_x)}_{i+\frac{1}{2},j} = \tilde{a}_{i+\frac{1}{2},j} \cdot \tilde{p}_{x i+\frac{1}{2},j} + c_a^{(1)} c_{p_x}^{(1)} \frac{h_y^2}{12} + o(h_y^4) \quad (22)$$

Thus the $\widetilde{(ap_x)}_{i+\frac{1}{2},j}$ depends on \bar{a} and \bar{p} in the rectangle of cells $(i-1, j-2) \rightarrow (i+2, j+2)$. Similar expressions can be derived for $\widetilde{(ap_x)}_{i-\frac{1}{2},j}$ as well as for the y -components $\widetilde{(ap_y)}_{i\pm\frac{1}{2},j}$. This results in a 25-point stencil that uses the $(i-2, j-2) \rightarrow (i+2, j+2)$ square around cell (i, j) .

This matrix constructed with the above stencils is completed with boundary conditions for \bar{a} and \bar{p} . For the density field, either Neumann or extrapolation boundary conditions are used depending on the specific physical domain configuration. For pressure, Neumann conditions are used for all cells on the computational domain boundaries, except the cell in the top right corner where pressure is fixed to zero.

The *hypre* package [28] is used to solve the linear system resulting from the above fourth-order discretization. The solution is based on a conjugate gradient method preconditioned with a 2nd order structured multigrid method to improve the convergence rate of the iterative method. About two iterations are necessary to reduce the residual of the linear system by one order of magnitude. The tolerance threshold for the residual is typically set to 10^{-14} – 10^{-17} to limit the propagation of convergence errors. The number of iterations necessary to achieve convergence is nearly independent of the problem size.

3.3. Computation of Pressure Gradient Correction

The cell-averaged pressure values computed by solving the above linear system are then used to compute the correction to be added to the provisional velocity field $\hat{\mathbf{v}}$. This is done in two steps:

1. The edge-averaged $\frac{1}{\rho}\widetilde{\nabla p}$ is computed using eqns. (18) and (22).
2. The edge-centered $\frac{1}{\rho}\nabla p$ can be computed using either explicit or implicit algorithms. The explicit algorithm relies on the power series expansion (19). The first coefficient in this expansion (20) corresponds to the edge center value. The x -direction correction for $u_{i+\frac{1}{2},j}$ can be written as follows, using $\xi \equiv p_x/\rho$ for notational compactness:

$$\xi_{i+\frac{1}{2},j} = \frac{9\tilde{\xi}_{j-2} - 116\tilde{\xi}_{j-1} + 2134\tilde{\xi}_j - 116\tilde{\xi}_{j+1} + 9\tilde{\xi}_{j+2}}{1920} \quad (23)$$

The expression for the y -direction corrections are similar, except $i \leftrightarrow j$.

The implicit algorithm relies on eq. (16) used to construct cell/edge-averages. The edge-average $\left(\frac{p_x}{\rho}\right)$ can be written, again using the above ξ , as:

$$\frac{-17\xi_{i+\frac{1}{2},j-2} + 308\xi_{i+\frac{1}{2},j-1} + 5178\xi_{i+\frac{1}{2},j} + 308\xi_{i+\frac{1}{2},j+1} - 17\xi_{i+\frac{1}{2},j+2}}{5760} = \tilde{\xi}_{i+\frac{1}{2},j} \quad (24)$$

This results in a pentadiagonal system for each $(i + \frac{1}{2})$ line in the y -direction to determine the corrections that need to be applied to the intermediate x -component of the velocity field. Similar linear systems are created for each $(j + \frac{1}{2})$ line to compute the corrections for the y -component of the velocity field.

Numerical tests for the computation of the pressure gradient correction show negligible differences between the values obtained with the explicit and implicit algorithms described above. In the numerical simulations we will use the explicit algorithm as it involves less operations and is simpler to implement in parallel than the implicit approach.

3.4. Boundary conditions

We discuss here the treatment of boundary conditions for the velocity and hydrodynamic pressure in Stages 1 and 3, and for temperature and species mass fractions in Stage 2.

Velocity and Hydrodynamic Pressure

Standard Dirichlet boundary conditions are imposed for the velocity components normal to inflow and symmetric boundaries. For components tangential to these types of boundaries, slip boundary conditions are applied through appropriate stencils used to compute the corresponding convective and diffusion fluxes.

A “convective” boundary condition is used for the velocity components normal to the outflow boundaries.

$$\frac{\partial (\mathbf{v} \cdot \mathbf{n})_b}{\partial t} + U_0 \frac{\partial (\mathbf{v} \cdot \mathbf{n})_b}{\partial n} = 0 \quad (25)$$

Here, subscript b refers to the boundary values, \mathbf{n} is the unit vector normal to the boundary pointing outside the computational domain and $\partial/\partial n$ is the partial derivative normal to the boundary. Outflow boundary conditions are commonly used in incompressible or low-Mach number flow computations to ensure that numerical errors near outflow boundaries are convected out of the computational domain.

The “convective” velocity U_0 is computed using a global mass conservation constraint obtained by integrating the continuity eq. (1a) over the computational domain.

$$\int_V \left(\nabla \cdot \mathbf{v} + \frac{1}{\rho} \frac{D\rho}{Dt} \right) dv = 0 \rightarrow U_0 = \pm \frac{1}{A_{out}} \left(\int_{A_{in}} (\mathbf{v} \cdot \mathbf{n}) d\sigma + \int_V \frac{1}{\rho} \frac{D\rho}{Dt} dv \right) \quad (26)$$

Here A_{in} and A_{out} are the areas of the inflow and outflow boundaries and the sign of the velocity depends on the direction of the unit normal \mathbf{n} . For Stage 1_a, U_0 is computed using $\left. \frac{1}{\rho} \frac{D\rho}{Dt} \right|^{n+1}$ extrapolated with eq. (9), while for Stage 3_a this term is computed using scalar values obtained in Stage 2.

The boundary conditions for the provisional velocity field are set similar to Kim and Moin [29] and the pressure equation discretization is adjusted accordingly on the boundaries of the computational domain. The pressure solve also requires global mass conservation to be satisfied to machine precision. In order to ensure this, eq. (1a) is again integrated over the computational domain after the boundary conditions are applied at the end of Stages 1_a and 3_a

$$\int_V \left(\nabla \cdot \mathbf{v} + \frac{1}{\rho} \frac{D\rho}{Dt} \right) dv = \pm \epsilon A_{out} \quad (27)$$

An average velocity correction ϵ is then added to the velocity components normal to the outflow

boundaries in order to ensure global mass conservation.

Temperature and Species Mass Fractions

For Stage 2_b , Dirichlet conditions are used for the scalar fields at the inflow boundaries. The Neumann conditions corresponding to symmetry boundary conditions are imposed through appropriate stencils in the calculation of convective and diffusion fluxes in eq. (1c-1d). A “convective” transport equation is used for scalars at outflow boundaries, similar to eq. (25) for the velocity field.

4. Results

As we developed the numerical construction presented in the previous section we used the method of manufactured solutions [30] to verify the implementation of various algorithm components. In this section we present results for one-dimensional (1D) and two-dimensional (2D) configurations that are used to test the stability and convergence rates for the full algorithm. First, the numerical stability and convergence rates are studied for a configuration that involves the advection-diffusion of a passive scalar. The passive scalar tests are used to extend the results established in [10] (for the correct pairings of the orders of interpolation and derivative stencils for nodal variables) to situations which involve a combination of cell-centered scalar variables and velocities on a staggered mesh. The tests also examine the efficiency of an RKC integrator for time advancement of discretized systems that include both convection and diffusion terms. Subsequently, we present numerical studies for full low Mach number reacting flow. Due to the computational expense of the reacting flow tests, time convergence rates are measured only in 1D configurations, while spatial convergence rates are measured both in 1D and 2D configurations. All reacting flow tests involve methane combustion, and chemistry is modeled using a C1 skeletal mechanisms with 16 species and 46 reversible reactions [31]. Transport properties are computed using a mixture-averaged formulation [25].

4.1. Convergence Tests Using an Advection-Diffusion System

For the simulations presented in this section, the velocity field is fixed and there is no scalar source term, therefore only Stage 2_b of the numerical algorithm is active. This series of tests involves the advection-diffusion of both a “plus”-like shape and a Gaussian pulse [32]. The first configuration is used to illustrate the lack of distortion in the convective transport of strong scalar gradients at off-axis directions through coarse-fine mesh boundaries, for sufficiently small convective CFL numbers.

The second configuration is used to illustrate the temporal and spatial convergence rates of the numerical construction.

A unit square $[0, 1] \times [0, 1]$ computational domain is considered and the advection corresponds to a solid body rotation around its center, $u = -\omega \left(y - \frac{1}{2}\right)$, $v = \omega \left(x - \frac{1}{2}\right)$. Since velocity is fixed and there are no reactions, for this series of test only Stage 2_b of the numerical algorithm is active.

In the first configuration the initial conditions are constructed as follows. First the lengths of the horizontal and vertical arms of the “plus” shape are set equal to 0.2 and their thicknesses to 0.05. These arms intersect symmetrically at $(x = 0.75, y = 0.5)$. The scalar field is then defined as

$$\phi_{\text{plus,IC}}(x, y, t) = \frac{1 + \tanh(\alpha G(x, y))}{2} \quad (28)$$

where $G(x, y)$ is the distance from point (x, y) to the closest point on the shape edge, positive inside the shape and negative outside it. Parameter α controls the slope of the scalar profile near the edges of the initial shape, and for these tests is set to $\alpha = 100$.

The initial and boundary conditions for the Gaussian pulse are computed according to the analytical solution

$$\phi_a(x, y, t) = \frac{\delta^2}{\delta^2 + 4\nu t} \exp\left(-\frac{(\bar{x} - x_0)^2 + (\bar{y} - y_0)^2}{\delta^2 + 4\nu t}\right) \quad (29)$$

where $\bar{x}(t) = \left(x - \frac{1}{2}\right) \cos(\omega t) + \left(y - \frac{1}{2}\right) \sin(\omega t)$, $\bar{y}(t) = -\left(x - \frac{1}{2}\right) \sin(\omega t) + \left(y - \frac{1}{2}\right) \cos(\omega t)$. The Gaussian pulse is initially located at $x_0 = 0.75$, $y_0 = 0.5$ and its initial diameter is set to $\delta = 0.045$.

Several values are considered for the diffusion coefficient ν and for the grid and time step sizes to determine the behavior of high-order stencils for prolongation and restriction and the convergence rates of the numerical construction. Figure 5 shows sample results for SAMR simulations using 2 mesh levels with 256×256 cells on the coarse grid. The finer mesh patches are stationary and placed such that scalar gradients cross the coarse-fine interfaces several times during each simulation. The results in the left frame are based on a diffusion coefficient $\nu = 10^{-4}$. The the three shapes correspond to the initial conditions, and after $3/8$ and $5/8$ time fractions of a full rotation period $t = \frac{2\pi}{\omega}$, respectively. The sample results in the right frame of Fig. 5 are obtained with a diffusion coefficient $\nu = 10^{-3}$. The snapshot corresponds to a $5/8$ time fraction of a full rotation period $t = 2\pi/\omega$. Comparison of contour levels on the coarse and fine mesh levels show no significant

distortion of the Gaussian pulse as it passes through them. A comparison with the analytical solution below provides a more rigorous check.

Figure 6 shows scalar contours for two simulations using the same grid size, 256^2 grid points, time step, $\Delta t = 10^{-4}$, and diffusion coefficient $\nu = 10^{-3}$. The diffusion CFL number, $\nu\Delta t/h_x^2$, which characterizes the discretization of the diffusion term, is the same for both simulations, 0.01. Here $h_x = h_y = 1/256$. The convection CFL number $\mathbf{u}\Delta t/h_x$ which is characteristic of the advection term discretization is 0.025 and 0.25, respectively. The solution for the latter, convection dominated flow, in the right frame, shows “wavy” contours indicating lack of stability of the time integration scheme at this convective CFL number. The solution in the left frame, where advection is much smaller, shows a “plus” shape free of instabilities.

Figure 7 shows results aimed at testing the behavior of the high-order spatial discretizations as the scalar gradients pass through coarse fine grid interfaces. In the left frame the coarse grid resolution leads to approximately 2-3 grid points inside the steep scalar gradient. This resolution is evidently not enough to resolve the spatial structure of the scalar field and the results obtained on a uniform mesh exhibit differences compared to the SAMR results. In the right frame, there are about 9 grid points in the regions where scalar values exhibit a large jump. For these tests there is good agreement between uniform and SAMR results.

The L_2 errors for several test simulations are shown in Fig. 8. These errors measure the difference between the numerical and analytical solutions:

$$L_2 = \sqrt{\frac{\sum_{i,j} (\phi_{i,j} - \phi_{i,j,a})^2}{N_p}} \quad (30)$$

where $\phi_{i,j}$ and $\phi_{i,j,a}$ are the numerical and analytical solutions, respectively, at grid cell (i, j) , and the errors are summed over all N_p grid cells. The analytical solution is given in eq. (29). In order to determine the convergence rates for the numerical integration, one needs to ensure that errors due to the spatial and temporal discretizations do not contaminate each other. The left frame shows L_2 errors measured after one full rotation for simulations using 1024^2 grid cells and time step values $\Delta t = 2.5 \times 10^{-5} \dots 2 \times 10^{-4}$. The L_2 error slopes for $\Delta t > 5 \times 10^{-5}$ agree well with the theoretical 2ndorder convergence rate of the RKC scheme. There is a slight degradation in the convergence rate for the simulation using $\nu = 10^{-3}$ and $\Delta t = 2.5 \times 10^{-5}$. For this simulation, the diffusion CFL

number is 0.025, which is small enough to allow contamination of the time discretization errors by spatial discretization errors, for sufficiently small Δt .

The right frame shows convergence results for simulations using $\Delta t = 2.5 \times 10^{-5}$ and grid sizes $N = 64 \dots 512$. Here N is the number of grid points in each computational direction. For these simulations the diffusion CFL number is 0.05 or less, which ensures that time discretization errors are small compared to spatial errors. The L_2 errors exhibit 4th-order convergence rates both for uniform mesh simulations (solid lines and circles) and SAMR simulations using two mesh levels (dashed lines and squares).

4.2. Temporal Convergence Tests Using Flame Simulations

For temporal convergence tests, the initial condition corresponds to a freely propagating premixed flame, computed with Chemkin's Premix package [33]. The computational domain is 1.5 cm long, and the premixed flame propagates from right to left into an unburnt stoichiometric mixture of methane (CH_4) and air. The initial solutions from the Chemkin package were relaxed on successively refined grids to ensure consistent initial conditions for the convergence tests. Both uniform and SAMR computational grids were used to determine the convergence rates of the numerical scheme. In the SAMR tests, three mesh levels were employed.

Figure 9 shows sample major species and radicals mass fractions profiles corresponding to the initial conditions for the multi-level convergence test. For the major species (left frame) the coarse grid solution is sufficient to capture their mass fraction profile through the flame. For some of the radicals however, e.g. HCO, the computational grid needed refinement in order to capture mass fraction profiles.

Time-dependent SAMR 1D flame computations were verified against corresponding 1D flame solutions using Chemkin Premix [33]. The resulting profiles of flame species in Fig 9 evolve in a self-similar manner in time as expected (not shown), and fall on top of the Chemkin initial condition solution when shifted back in time.

We focus in the following on self-convergence studies with the present SAMR construction. Tables 1 and 2 show select results for time convergence tests using a uniform mesh and a 3-level mesh, respectively. Since for these simulations there is no analytical solution, the errors are computed by comparing the solutions obtained with successively refined time steps. For the tests using a

uniform mesh, $M = 16$ RKC stages were required to have stable simulations with the largest time step $\Delta t = 5 \times 10^{-7}$ s. For the multi-level mesh, $M = 8$ RKC stages were sufficient to ensure stable simulations since the grid size on the coarse mesh level was 4 times larger compared to the uniform mesh simulation. Simulations were advanced in time for 0.4 ms. During this interval the flame advanced about 0.2 mm, or about half its width, into the fresh mixture. These tests show 2nd order temporal convergence for all variables.

4.3. Spatial Convergence Tests Using Flame Simulations

In this section we investigate the performance of our construction focusing on spatial accuracy and robustness. Primarily, we seek to illustrate that the full coupled construction has 4th order accuracy. Secondly, we investigate whether the rules for pairing interpolation and derivative stencils, derived for nodal variables in [10], hold for a mixture of edge- and cell-centered variables. In this respect, 2D simulations, which involve a large number of interpolations of velocities and scalar variables away from their collocation points, pose a stringent test of their accuracy and stability.

4.3.1. 1D Freely Propagating Premixed Flame

The spatial convergence rates were first computed using the same configuration as for the time convergence tests shown in Section 4.2. For this series of tests, the Chemkin solution is first relaxed on the mesh with the finest grid size. The initial conditions for the coarser computational grids are then obtained by interpolation using 6th order stencil derived in Appendix B.

Figure 10 shows convergence rates obtained from simulations using 3 mesh levels, coarse grid sizes from $7.5\mu\text{m}$ to $30\mu\text{m}$, and a time step 10^{-8} s. Similar to the time convergence tests, the spatial errors were assessed by comparing the solutions on successively smaller grid sizes. The L_2 errors in Fig. 10 exhibit a 4th-order decay with the grid size.

4.3.2. 2D Vortex Pair - Flame Interaction

A canonical vortex-flame configuration [2] was chosen to explore the performance of the numerical construction. Figure 11 shows a schematic of the computational domain, which is (1.5×0.75) cm in the streamwise and transverse directions, respectively. The velocity field corresponding to a periodic row of counter-rotating Lamb-Oseen vortices is superimposed over the premixed 1D flame solution discussed above. The methodology is described below.

The azimuthal velocities induced by the positive and negative components of the vortex pair are computed as

$$V_{\theta,\pm}(x, y) = \pm \frac{\Gamma_0}{2\pi r_{\pm}} \left(1 - \exp\left(-\frac{r_{\pm}^2}{4\nu t_0}\right) \right) \quad (31)$$

where r_+ and r_- are the distances from a given point (x, y) to the centers of the positive (x_+, y_+) and negative (x_-, y_-) vortices, respectively. The vortices are placed symmetrically with respect to the streamwise centerline:

$$x_{\pm} = \frac{W}{2} \pm x_{vp}; \quad y_{\pm} = y_{vp} \quad (32)$$

where $W = 0.75$ cm is the width of the computational domain. The u and v components of the velocity field are then computed as

$$\begin{aligned} u(x, y) &= -(V_{\theta,+}(x, y) \frac{y - y_+}{r_+} + V_{\theta,-}(x, y) \frac{y - y_-}{r_-}) \\ v(x, y) &= V_{\theta,+}(x, y) \frac{x - x_+}{r_+} + V_{\theta,-}(x, y) \frac{x - x_-}{r_-} \end{aligned} \quad (33)$$

In order to simulate the effect of periodic vortex pairs, contributions from virtual pairs to the left and right of the computational domain need to be added to the velocity field. Their effect decreases exponentially with the distance and we found that 16 pairs on each side of the domain are sufficient to generate accurate initial conditions for the velocity field. The parameters for the Lamb-Oseen vortices are as follows: $\Gamma_0 = 0.04$ m²/s, $\nu = 1.568 \times 10^{-5}$ m²/s, $t_0 = 5 \times 10^{-3}$ s, and $(x_{vp}, y_{vp}) = (0.05, 0.6)$ cm.

Throughout the simulation, the velocity at the inflow boundary is kept constant, while the velocity at the outflow boundary is determined based on mass conservation constraints. Symmetry boundary conditions are imposed on the lateral boundaries.

Numerical simulations were performed using both uniform and SAMR grids with 2 and 3 mesh levels. Three coarse grid sizes were considered, $h_x = 15\mu\text{m} \dots 60\mu\text{m}$, to determine spatial convergence rates. A time step $\Delta t = 2 \times 10^{-8}$ s was adopted for all simulations. This time step leads to a convection CFL number of about 0.05 and a diffusion CFL number based on H₂ diffusivity of about 0.3. This time step ensures that the stability region of the RKC scheme can handle convection and diffusion terms simultaneously, as observed in Section 4.1.

Figure 12 shows freeze frames of the vorticity field and flame heat release rate. The vortex

pair is initially located 2 mm upstream of the flame and propagates with a speed of approximately 8 m/s towards it. As the vortex pair impinges into the flame, the flame structure is modified by the interaction with the strain rate field induced by the vortex pair and, at later times, the flame is quenched on the centerline (results not shown) by the vortex pair. **Flame structure and dynamics during the interaction with the vortex pair are consistent with results in earlier low-Mach number reacting flow studies focusing on flame-vortex interactions [34].**

Figure 13 shows the x -velocity, vorticity, and heat release rate (HRR) profiles along centerline and a slanted cut through the flame. The circles correspond to a uniform mesh simulation with $h_x = 15\mu\text{m}$ while the square symbols correspond to a 2 level SAMR simulation with $h_x = 30\mu\text{m}$ on the coarse level. The velocity and vorticity profiles for the SAMR simulation correspond to the coarse level, while the heat release rate profile for the SAMR correspond to the fine level. The velocity field is captured well by the SAMR simulation on the coarse grid compared to the uniform grid simulation. The HRR results are also in very good agreement.

Figure 14 shows details of the velocity field and HCO mass fraction profiles in the flame region. These results show that the inner flame structure is much thinner than the velocity field length scales in the flame region and elsewhere in the vortex pair region. The good agreement between the uniform mesh and SAMR simulations indicates that mesh refinement is only necessary for resolving inner flame structure, while the spatial length scales associated with the flow need to be captured on the lowest level of the SAMR hierarchy only.

These results in Fig. 15 are used to further test the effect of coupling the momentum transport, solved on the coarsest level of the mesh hierarchy, with the scalar transport, solved throughout the mesh hierarchy. This figure shows the pressure and HCO mass fraction fields for a uniform mesh simulation with $h_x = 15\mu\text{m}$ and a three level SAMR simulation. The finest level on the SAMR simulation has the same grid size as the uniform mesh simulation, while its coarsest level has a resolution $h_x = 60\mu\text{m}$ that is four times larger. For the SAMR simulation, the pressure field is shown on the base level in the grid hierarchy, while the mass fraction fields are shown on the finest level. The results for the two simulations are in very good agreement, showing that mesh refinement is only necessary to solve the inner flame structure as long as coarsest mesh level is sufficient to capture the velocity field dynamics.

The results in Table 3 show the spatial convergence rates for this configuration, corresponding

to the middle frame in Fig. 12, or about 0.3 ms from the start of the simulation. Roughly 4th-order convergence rate is observed for all variables.

4.4. Computational Expense of the Algorithm Components

Several flame-vortex interaction simulations, referred here as *R1* through *R6*, were performed to analyze the computational expense of various algorithm components. The computational domain extent for these simulations is the same as in Section 4.3.2. Runs *R1* and *R2* are uniform mesh simulations with 1024×512 and 512×256 grid points, respectively. Runs *R3* through *R6* are SAMR simulations with one level of refinement, with the same mesh resolution on the coarse/fine level as run *R2/R1*. The fine levels for these runs cover 5%, 7%, 12%, and 30%, respectively, of the coarse level area.

We implement a time step ratio of 1/2 between successive mesh levels, *i.e.* $\Delta t_{L+1} = \Delta t_L/2$, with the time step size on the finest mesh level chosen according to RKC scalar diffusion stability considerations. For *R2* and the SAMR simulations *R3* through *R6* this results in a time step twice larger compared to the uniform mesh simulation *R1*. All these runs were advanced for 0.02 ms to test the computational gains for SAMR. The relative CPU times for the SAMR runs were between 17% – 37% of the computational cost for *R1*, as shown in Table 4.

Figure 16 shows relative CPU times for runs *R2* through *R6*. For all runs, the computational expense of Stages 1_a , 1_c , 3_a , and 3_c is negligible. Both Stages 1_b and 3_b involve an elliptic solve for the pressure field. Most of the computational time is spent in the computation of $\frac{1}{\rho} \frac{D\rho}{Dt}$ during Stage 3_b , while for Stage 1_b this term is extrapolated from values at previous time steps. The elliptic system is solved using the multigrid preconditioners provided by *hypre* library, and the computational cost is small compared with the rest of the algorithm.

Stage 2 consumes the majority of the computational time. For a uniform grid simulation (*R2*) the time integration of the chemical source term, in Stages 2_a and 2_c , is nearly equal with the cost of the RKC scheme for the scalar convection and diffusion terms in Stage 2_b . Runs *R3* through *R6* show an increase in the relative cost of Stage 2_b as the refinement area increases. This is due to the fact that the RKC time integration involves prolongations and restrictions between various grid levels as well as data exchange between grid patches on the same level. In contrast, Stages 2_a and 2_c require very little data exchange between the computational grid levels.

The relative expense of Stages 2_a and 2_c is expected to increase for more detailed and stiff kinetic models, compared to the current mechanism. However, the above results suggest that using shallower grid hierarchies, made possible by high-order discretizations, will allow more efficient computations through a reduction of data transferred between and across various levels in the grid hierarchy.

5. Conclusions

This paper introduces a high-order numerical model for the simulation of chemically reacting flow in the low-Mach number limit. A 4th order (in space) projection algorithm for the momentum transport is coupled to a 4th order-in-space, 2nd order-in-time scheme for the solution of the equations of transport of energy and species mass fractions on a block-structured adaptively refined mesh. Finite differences are used to approximate spatial derivatives.

We develop here high-order stencils for derivatives and interpolation of cell-centered and edge-centered variables. The correct pairings of order-of-accuracy for derivatives and interpolations, originally derived for nodal variables in [10], were verified and employed for the new cell- and edge-centered stencils. An extended stability explicit integrator, RKC, originally employed in an SAMR configuration for time advancement of a reactive-diffusive system [9], is used here for a flame simulation employing the low Mach number approximation of the Navier-Stokes equation. The RKC time-integrator is coupled to a projection scheme to solve the resulting system of differential-algebraic equations. The projection scheme for the momentum solution adapts a finite-volume construction from [27] to finite differences, in order to achieve a consistent fourth-order construction for the pressure (Poisson) solve. The variable coefficient Poisson problem is solved using a conjugate-gradient method, preconditioned with a multigrid technique in the *hypr*e package [28] to accelerate the convergence rate.

Canonical advection-diffusion systems and vortex-flame interactions are used to investigate the performance of the numerical construction. In the advection-diffusion system, analytical shapes cross several refined grids in a 2D SAMR hierarchy. This test is used to test the efficiency and stability of the interpolation and derivative stencils as the scalar structure crosses the coarse-fine grid interfaces. The theoretical convergence rates, 2nd order in time and 4th order in space, are observed by comparing the numerical and analytical solutions.

Both 1D and 2D computational domains were used in the flame simulations. These tests involve methane combustion modeled using a skeletal C1 mechanism. In the 1D configuration, a premixed flame propagates into a quiescent fresh mixture of fuel and oxidizer, while in the 2D configuration a vortex pair propels itself into an initially flat premixed flame. The temporal and spatial convergence rates were confirmed by comparing the solutions obtained with several grid and time step sizes. The 2D configuration is also used to evaluate our approach of solving for the flow field on the coarse level only, while using mesh refinement to resolve the flame structure. For the configurations of interest, flow length scales are several times larger compared to finest length scales observed in the scalar fields. We resolve the velocity field on the lowest level of the SAMR hierarchy and interpolate it to finer levels to compute the convective term when time-advancing the scalars. This approach requires an elliptic pressure solve, necessary for the projection scheme, on the coarse level only, making the projection scheme simple to implement and inexpensive compared with the computational effort spent in time-advancing the scalars.

The primary reason for constructing a 4th-order SAMR scheme was to reduce the number of cells in the entire problem, prior to using it with detailed (and stiff) kinetic mechanisms in reacting flow studies. Based on current results for the computational expense of various algorithm components, chemistry integration costs are expected to dominate and efficiency/usability gains are tied to using shallow grid hierarchies and to keeping the number of cells at a “manageable” level, while resolving the flame structure adequately. Future efforts will be devoted to such problems and will necessarily exploit massive parallelism in computations.

Acknowledgments

This work was supported by the US Department of Energy (DOE), Office of Basic Energy Sciences (BES), SciDAC Computational Chemistry program. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000.

- [1] A. Majda, J. Sethian, The derivation and numerical solution of the equations for zero mach number combustion, *Comb. Sci. and Technology* 42 (1985) 185–205.

- [2] H. Najm, O. Knio, Modeling Low Mach Number Reacting Flow with Detailed Chemistry and Transport, *J. Sci. Comp.* 25 (1) (2005) 263–287.
- [3] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64–84.
- [4] B. Bennett, M. Smooke, Local rectangular refinement with application to axisymmetric laminar flames, *Combust. Theor. Model.* 2 (3) (1998) 221–258.
- [5] J. Ray, H. Najm, R. Milne, K. Devine, S. Kempa, Triple flame structure and dynamics at the stabilization point of an unsteady lifted jet diffusion flame, *Proc. Combust. Inst.* 28 (2000) 219–226, part 1.
- [6] M. Day, J. Bell, Numerical simulation of laminar reacting flows with complex chemistry, *Combust. Theor. Model.* 4 (4) (2000) 535–556.
- [7] M. Anthonissen, B. Bennett, M. Smooke, An adaptive multilevel local defect correction technique with application to combustion, *Combust. Theor. Model.* 9 (2) (2005) 273–299.
- [8] J. G. Verwer, B. Sommeijer, W. Hundsdorfer, RKC Time-Stepping for Advection-Diffusion-Reaction Problems, *J. Comput. Phys.* 201 (1) (2004) 61–79.
- [9] S. Lefantzi, J. Ray, C. Kennedy, H. Najm, A component-based toolkit for simulating reacting flows with high order spatial discretisations on structured adaptively refined meshes, *Prog. Comput. Fluid Dyn.* 5 (6) (2005) 298–315.
- [10] J. Ray, C. Kennedy, S. Lefantzi, H. Najm, Using High-Order Methods on Adaptively Refined Block-Structured Meshes, *SIAM J. Sci. Comp.* 29 (1) (2007) 139–181.
- [11] A. Almgren, J. Bell, P. Colella, L. Howell, M. Welcome, A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier-Stokes Equations, *J. Comput. Phys.* 142 (1998) 1–46.
- [12] D. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible euler equations, *J. Comp. Phys.* 163 (2000) 271–312.

- [13] H. Dimmelmeier, J. Novak, J. A. Font, J. M. Ibáñez, E. Müller, Combining spectral and shock-capturing methods: A new numerical approach for 3d relativistic core collapse simulations, *Phys. Rev. D* 71 (6) (2005) 064023.
- [14] M. D. Duez, F. Foucart, L. E. Kidder, H. P. Pfeiffer, M. A. Scheel, S. A. Teukolsky, Evolving black hole-neutron star binaries in general relativity using pseudospectral and finite difference methods, *Physical Review D (Particles, Fields, Gravitation, and Cosmology)* 78 (10) (2008) 104015.
- [15] M. Barad, P. Colella, A fourth-order accurate local refinement method for Poisson’s equation, *J. Comput. Phys.* 209 (2005) 1–18.
- [16] X. Huang, X. Zhang, Block-structured adaptive mesh refinement for computational aeroacoustics, *Int. J. Comput. Fld. D.* Under review.
- [17] M. Day, J. Bell, Numerical simulation of laminar reacting flows with complex chemistry, *Combust. Theory Modelling* 4 (2000) 535–556.
- [18] J. Bell, M. Day, I. Shepherd, M. Johnson, R. Cheng, J. Grcar, V. Beckner, M. Lijewski, Numerical simulation of a laboratory-scale turbulent v-flame, *P. Natl. Acad. Sci. USA* 102 (29) (2005) 10006–10011.
- [19] J. Bell, M. Day, J. Grcar, M. Lijewski, D. JF, S. Filatyev, Numerical simulation of a laboratory-scale turbulent slot flame, *Proc. Combust. Inst.* 31 (2007) 1299–1307.
- [20] P. Colella, M. D. Sekora, Short note: A limiter for PPM that preserves accuracy at smooth extrema, *J. Comput. Phys.* 227 (15) (2008) 7069–7076.
- [21] P. Colella, M. Dorr, J. Hittinger, D. F. Martin, P. McCorquodale, High-order finite-volume adaptive methods on locally rectangular grids, *J. Phys. Conf. Ser.* 180 (2009) 012010.
- [22] X. Huang, X. Zhang, S. K. Richards, Adaptive mesh refinement computation of acoustic radiation from an engine intake, *Aerosp. Sci. Technol.* 12 (5) (2008) 418 – 426.
- [23] H. N. Najm, J. Ray, M. Valorani, F. Creta, D. A. Goussis, A computational facility for reacting flow science, *J. Phys. Conf. Ser.* 46 (2006) 53–57.

- [24] B. J. McBride, S. Gordon, M. A. Reno, Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species, Tech. Rep. NASA TM-4513, NASA (1993).
- [25] P. H. Paul, DRFM: A New Package for the Evaluation of Gas-Phase-Transport Properties, Sandia Report SAND98-8203, Sandia National Laboratories, Albuquerque, New Mexico (November 1997).
- [26] S. D. Cohen, A. C. Hindmarsh, CVODE, a Stiff/Nonstiff ODE Solver in C, *Comput. Phys.* 10 (2) (1996) 138–143.
- [27] S. Kadioglu, R. Klein, M. Minion, A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics, *J. Comput. Phys.* 227 (2008) 2012–2043.
- [28] hypre: High performance preconditioners, <http://computation.llnl.gov/casc/hypre>.
- [29] J. Kim, P. Moin, Application of a fractional-step method to incompressible navier-stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.
- [30] P. Roache, Code Verification by the Method of Manufactured Solutions, *ASME J. Fluids Engineering* 124 (2002) 4–10.
- [31] M. Smooke, V. Giovangigli, Reduced kinetic mechanisms and asymptotic approximations for methane-air flames, Vol. 384 of *Lecture Notes in Physics*, Springer-Verlag, Berlin, 1991, Ch. 1.
- [32] A. Rößler, M. Seaïd, M. Zahri, Method of lines for stochastic boundary-value problems with additive noise, *Appl. Math. Comput.* 199 (2008) 301–314.
- [33] R. Kee, J. Grcar, M. Smooke, J. Miller, A Fortran Program for Modeling Steady Laminar One-Dimensional Premixed Flames, Sandia Report SAND85-8240, Sandia National Labs., Livermore, CA (September 1993).
- [34] H. Najm, P. Wyckoff, Premixed Flame Response to Unsteady Strain-Rate and Curvature, *Combustion and Flame* 110 (1-2) (1997) 92–112.

Appendix A: Finite-Difference Discretization of the Pressure Equation

We discuss here the derivation of fourth-order finite difference discretization for the variable coefficient Poisson equation (8) for the pressure field. Consider, for simplicity, a one-dimensional configuration, with the indices for the cell and edge centers as shown in Fig. 17. The pressure gradient, $\frac{\partial p}{\partial x}$, is first computed at face centers using a 4th order approximation:

$$\left. \frac{\partial p}{\partial x} \right|_{i+\frac{1}{2}} = \frac{1}{24h_x} (p_{i-1} - 27p_i + 27p_{i+1} - p_{i+2}) \quad (34)$$

Density is interpolated from cell centers to edge centers using a 6th order stencil [10]. The 4th order derivative stencil is applied again to the product $\frac{1}{\rho} \frac{\partial p}{\partial x}$ and results in:

$$\left. \frac{\partial}{\partial x} \left(\frac{1}{\rho} \frac{\partial p}{\partial x} \right) \right|_i = \frac{1}{(24h_x)^2} (ap_{i-3} + bp_{i-2} + cp_{i-1} - (a + \dots + f)p_i + dp_{i+1} + ep_{i+2} + fp_{i+3}) \quad (35)$$

where

$$\begin{aligned} a &= \frac{1}{\rho_{i-\frac{3}{2}}}; & b &= - \left(\frac{27}{\rho_{i-\frac{3}{2}}} + \frac{27}{\rho_{i-\frac{1}{2}}} \right); & c &= \frac{27}{\rho_{i-\frac{3}{2}}} + \frac{27^2}{\rho_{i-\frac{1}{2}}} + \frac{27}{\rho_{i+\frac{1}{2}}} \\ f &= \frac{1}{\rho_{i+\frac{3}{2}}}; & e &= - \left(\frac{27}{\rho_{i+\frac{3}{2}}} + \frac{27}{\rho_{i+\frac{1}{2}}} \right); & d &= \frac{27}{\rho_{i+\frac{3}{2}}} + \frac{27^2}{\rho_{i+\frac{1}{2}}} + \frac{27}{\rho_{i-\frac{1}{2}}} \end{aligned} \quad (36)$$

A three-cell wide layer is constructed around the computational domain in order to impose boundary conditions and maintain 4th order accuracy. The pressure gradient is set to zero at the computational domain boundaries to maintain consistency with the boundary conditions for the velocity field. Figure 17 shows a detail perpendicular to the x_{\min} boundary. The cell center values in the outer layer are computed based on a fourth order discretization for the pressure gradient on the boundary and 5th order extrapolations for cells (-2) and (-3), namely:

$$\begin{aligned} \frac{1}{24h_x} (p_{-2} - 27p_{-1} + 27p_0 - p_1) &= 0 \\ p_{-2} - 5p_{-1} + 10p_0 - 10p_1 + 5p_2 - p_3 &= 0 \\ p_{-3} - 5p_{-2} + 10p_{-1} - 10p_0 + 5p_1 - p_2 &= 0 \end{aligned} \quad (37)$$

leading to

$$\begin{aligned}
p_{-3} &= \frac{1}{22} (-625p_0 + 1125p_1 - 603p_2 + 125p_3) \\
p_{-2} &= \frac{1}{22} (-135p_0 + 265p_1 - 135p_2 + 27p_3) \\
p_{-1} &= \frac{1}{22} (17p_0 + 9p_1 - 5p_2 + p_3)
\end{aligned} \tag{38}$$

These values are used to construct the stencils for the grid cell adjacent to the boundary. The stencil's orders for the face-centered densities near the boundary are reduced compared to the interior stencil to limit numerical instabilities associated with wide one-sided stencils. The face-centered densities $\rho_{3/2}$ and $\rho_{1/2}$ are computed using 5th and 4th order stencils, respectively. The values at $(-1/2)$ and $(-3/2)$ are extrapolated using 3rd order stencils.

$$\begin{aligned}
\rho_{3/2} &= \frac{1}{128} (-5\rho_0 + 60\rho_1 + 90\rho_2 - 20\rho_3 + 3\rho_4) \\
\rho_{1/2} &= \frac{1}{16} (5\rho_0 + 15\rho_1 - 5\rho_2 + \rho_3) \\
\rho_{-1/2} &= \frac{1}{16} (35\rho_0 - 35\rho_1 + 21\rho_2 - 5\rho_3) \\
\rho_{-3/2} &= \frac{1}{16} (105\rho_0 - 189\rho_1 + 135\rho_2 - 35\rho_3)
\end{aligned} \tag{39}$$

The above discretization results in a singular heptadiagonal linear system $Ap = b$. In order to ensure its consistency, it is necessary that the sum of the rhs term projected on the left-null space of A be identically 0. Specifically, let x be a left-null vector of A , $x^T \cdot A = 0$, for example computed using a singular value decomposition (*SVD*) algorithm. The system $Ap = b$ is consistent if $x^T b = 0$. The elements of b are a discretized version of the continuity equation (1a) using the provisional velocity field $\hat{\mathbf{v}}$. Integrating this equation over the computational domain and making use of the boundary conditions for the provisional velocity field $\hat{\mathbf{v}}$ as outlined in Section 3.4 we obtain

$$\int_V \left(\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \hat{\mathbf{v}} \right) dv + \int_A (\mathbf{v} - \hat{\mathbf{v}}) \cdot \mathbf{n} d\sigma = \int_V \frac{1}{\rho} \frac{D\rho}{Dt} dv + \int_A \hat{\mathbf{v}} \cdot \mathbf{n} d\sigma + \int_A (\mathbf{v} - \hat{\mathbf{v}}) \cdot \mathbf{n} d\sigma \tag{40}$$

$$= \int_V \frac{1}{\rho} \frac{D\rho}{Dt} dv + \int_A \mathbf{v} \cdot \mathbf{n} d\sigma = 0 \tag{41}$$

Therefore, the boundary conditions for the velocity field need to take into account the left null

vector x to ensure $x^T b = 0$. Since matrix A changes during the simulation as the density field evolves in time, the SVD procedure will need to be applied each time step in order to compute the required discretization for the boundary conditions applied to the velocity field. For 2nd order discretizations of the variable coefficient Poisson equation (8) the above procedure is not necessary since the left-null vector components are all equal to 1.

In order to avoid the costly procedure above, we implemented in Section 3.2 a discretization that effectively transforms the left null vector x to unit entries while retaining the 4th order accuracy for the pressure field.

Appendix B: Cell-centered Stencils for Interpolation and Derivatives

Interpolants

Figure 18 shows a schematic of 1D grid cells on adjacent coarse and fine grid levels. In this figure cell i_1 on the coarse level corresponds to the same computational space as cells $i_2 - 1$ and i_2 on the fine level. The 4th and 6th order interpolation stencils used to interpolate data between coarse and fine grid levels are provided below.

A N_i^{th} order, with N_i even, interpolation stencil between coarse f_c and fine f_f cell-centered values can be written as

$$\begin{aligned} f_{f,i_2} &= \sum_{j=-N_i/2+1}^{N_i/2} c_j f_{c,i_1+j} \\ f_{f,i_2+1} &= \sum_{j=-N_i/2+1}^{N_i/2} c_{1-j} f_{c,i_1+j} \end{aligned} \quad (42)$$

The coefficients for 4th and 6th order interpolations are given in Table 4.

Skewed one-sided interpolation stencils are used for computational cells near the boundary. In general skewed stencils are wider compared to their centered counterparts in order to achieve the same discretization order. In order to avoid numerical instabilities, their order is reduced compared to the stencils used for the interior grid cells. For N_i^{th} order interior stencils, the order of the

boundary stencils is $N_i - 1$:

$$f_{f,k} = \sum_{j=0}^{N_i-1} c_{k,j} f_{c,j} \quad k = 0, 1, 2, \dots \quad (43)$$

In the expressions above, k is the index for fine grid cells and j for the coarse grid cells. The cell index abutting the boundary has index 0 for both the coarse and the fine mesh levels. Coefficients $c_{k,j}$ are given in Tables 5 and 6 for $N_i = 4$ and $N_i = 6$, respectively.

The stencils for fine-to-coarse cell centered interpolations are similar to the ones derived for coarse-to-fine vertex centered interpolations in [10]. For cell centered interpolations in 2D and 3D configurations (see also Figs. 1 and 2) the one-dimensional stencils are applied successively in each coordinate direction.

Interpolations of edge-centered variables, e.g. velocity components, make use of both one-dimensional cell-centered presented above and vertex-centered stencils [10]. For example, for the x -velocity in a 2D configuration, in Fig. 2, the coarse field is first interpolated in the y -direction using 1D cell-centered interpolation stencils. The resulting field is then interpolated in the x -direction using vertex-centered stencils to obtain the x -velocity on a refined mesh level. Methodologies for other velocity components as well as for 3D configurations are similar.

Derivatives

An N_i^{th} order, with N_i even, staggered derivative at edge centers in the x -direction (see Fig. 1), denoted by fractional indices $i + \frac{1}{2}$, based on field values at cell centers, denoted by integer indices, can be written as

$$h_x \times \left. \frac{\partial f}{\partial x} \right|_{i+\frac{1}{2}} \approx \sum_{k=1}^{N_i/2} s_k (f_{i+k} - f_{i-k+1}) \quad (44)$$

The 4th order stencil coefficients are $s_1 = \frac{27}{24}$ and $s_2 = -\frac{1}{24}$. At locations near the domain boundaries, skewed stencils are used. These stencils are set to a lower order compared to their counterparts for the interior derivatives to avoid numerical instabilities. Assuming $i = 0$ is the index of the first cell

center in the computational x -direction, these derivatives are computed as

$$\begin{aligned}
 h_x \times \left. \frac{\partial f}{\partial x} \right|_{-\frac{1}{2}} &\approx -2f_0 + 3f_1 - f_2 && (2^{\text{nd}} \text{ order}) \\
 24h_x \times \left. \frac{\partial f}{\partial x} \right|_{\frac{1}{2}} &\approx -23f_0 + 21f_1 + 3f_2 - f_3 && (3^{\text{rd}} \text{ order})
 \end{aligned} \tag{45}$$

Derivatives in other computational directions are similar. These stencils are also used to compute derivatives of edge centered variables computed at either cell centers, e.g. in Stage 2_b , or at cell-vertices, e.g. off-diagonal stress tensor components in Stage 1_a , see Section 3.1.

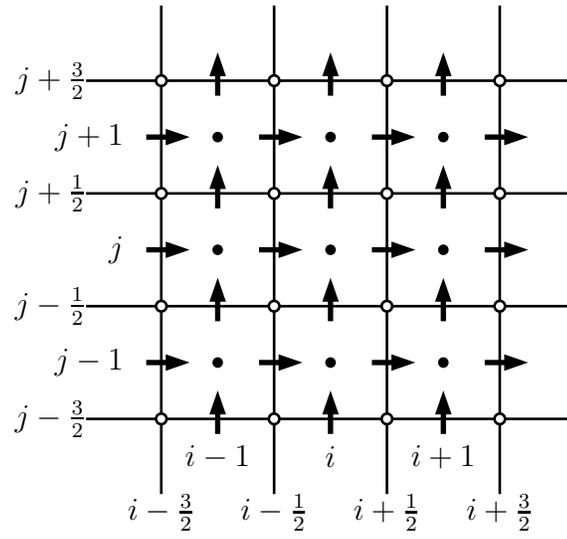


Figure 1: Staggered grid indices and variable locations.

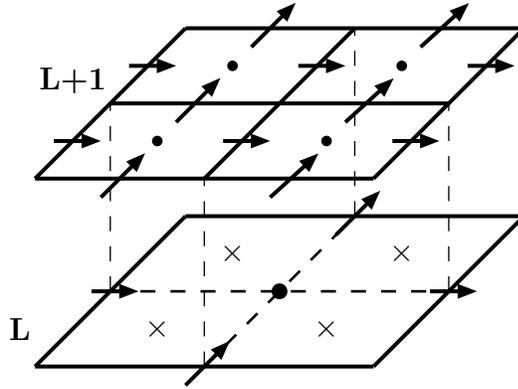


Figure 2: Cell topology on consecutive grid levels, L and $L + 1$. The cell centers are shown with filled symbols and the \times symbols correspond to the coarse grid (L) location of the fine grid ($L + 1$) cell-centers. The velocity field components are shown with arrows.

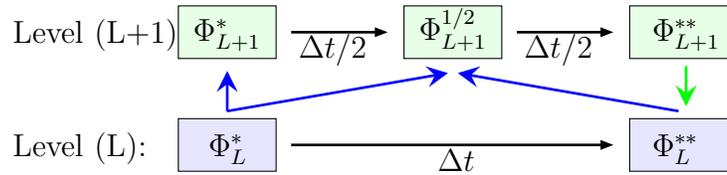


Figure 3: Schematic for Berger-Collela time advancement for scalars.

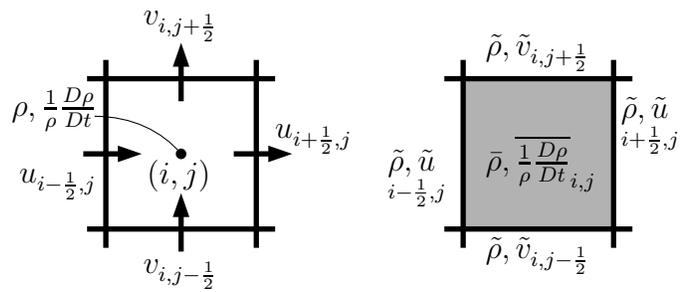


Figure 4: Schematic of the correspondence between cell- and edge-centered variables (left side) and their cell- and edge-averaged counterparts (right side). The bar denotes cell-averages and the tilde edge-averages.

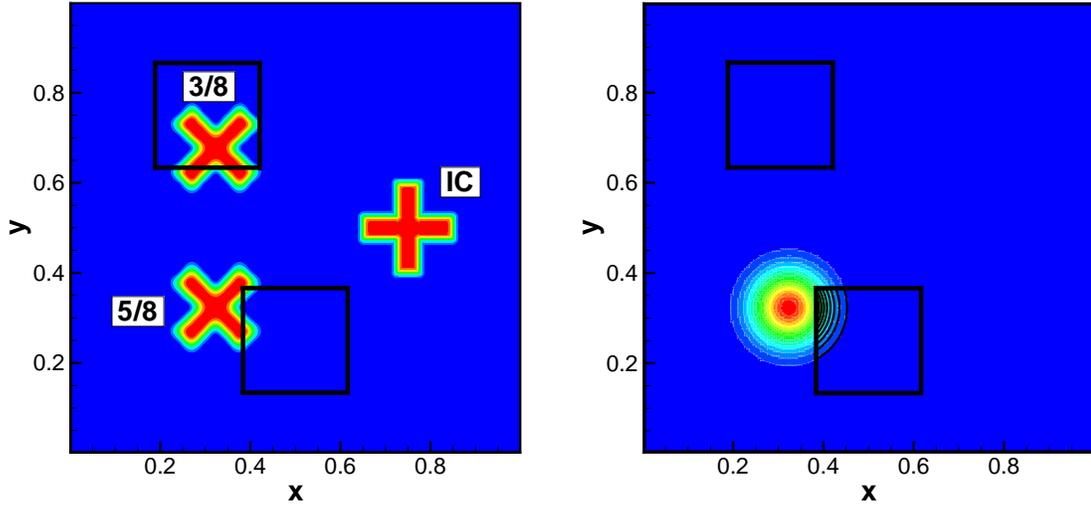


Figure 5: Left frame: Advection-diffusion of passive scalar based on a “plus” shape. Shaded contours show the solution on the base level. Right frame: Contour lines of passive scalar Gaussian pulse. The shaded contours correspond to the solution of the coarse mesh level, and the black contours correspond to the finer mesh level. Both simulations use a coarse grid with 256^2 grid cells and two refined stationary blocks.

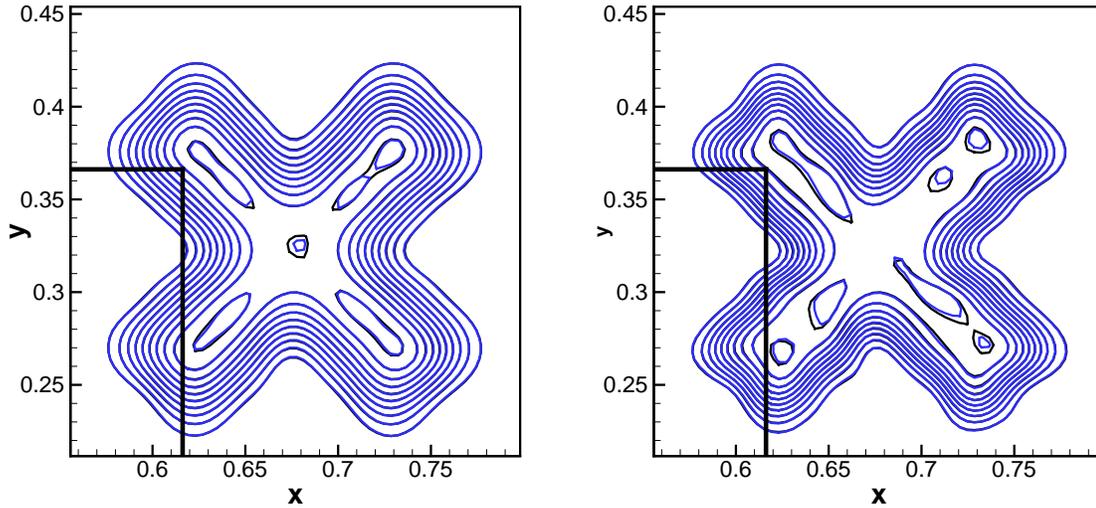


Figure 6: Passive scalar contours for SAMR simulations using 1 and 2 mesh levels. The solutions are shown at a $7/8$ time fraction of a full rotation. Black contours correspond to the coarse level runs and blue contours to the coarse+fine level runs. The coarse grid resolution and fine level blocks are the same as in Fig. 5. The left frame solution corresponds to a convective $CFL=0.025$ while for the right frame $CFL=0.25$. The diffusive CFL is the same for both simulations, 0.01 .

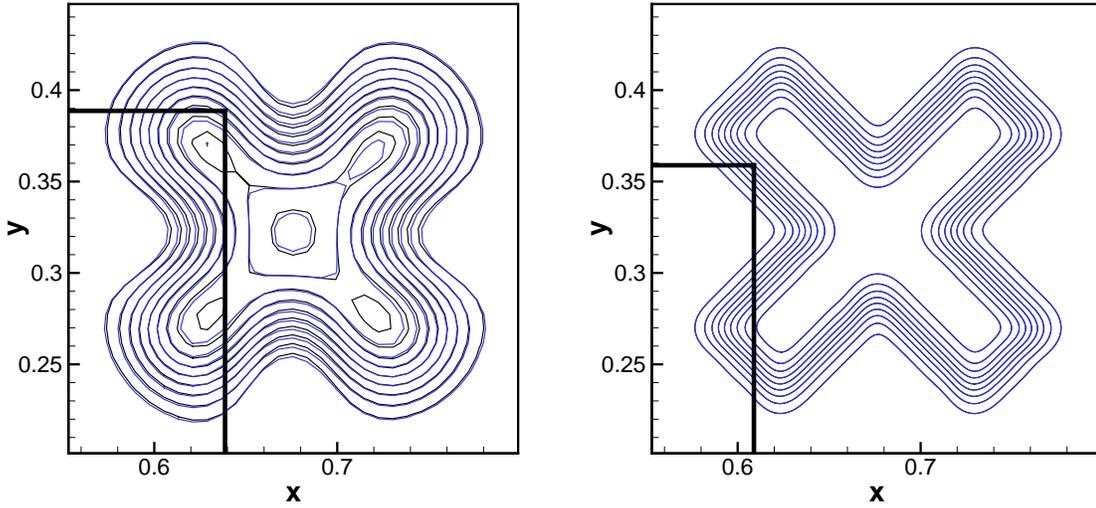


Figure 7: Passive scalar contours for simulations using uniform and SAMR with 2 mesh levels. The solutions are shown at a $7/8$ time fraction of a full rotation. Black contours correspond to the uniform mesh simulations and blue contours to the coarse level of the SAMR simulations. The left frame solution corresponds to a 128^2 grid points on the coarse level, while the right frame to 528^2 grid points. The uniform mesh and coarse level resolutions are the same in the corresponding simulations.

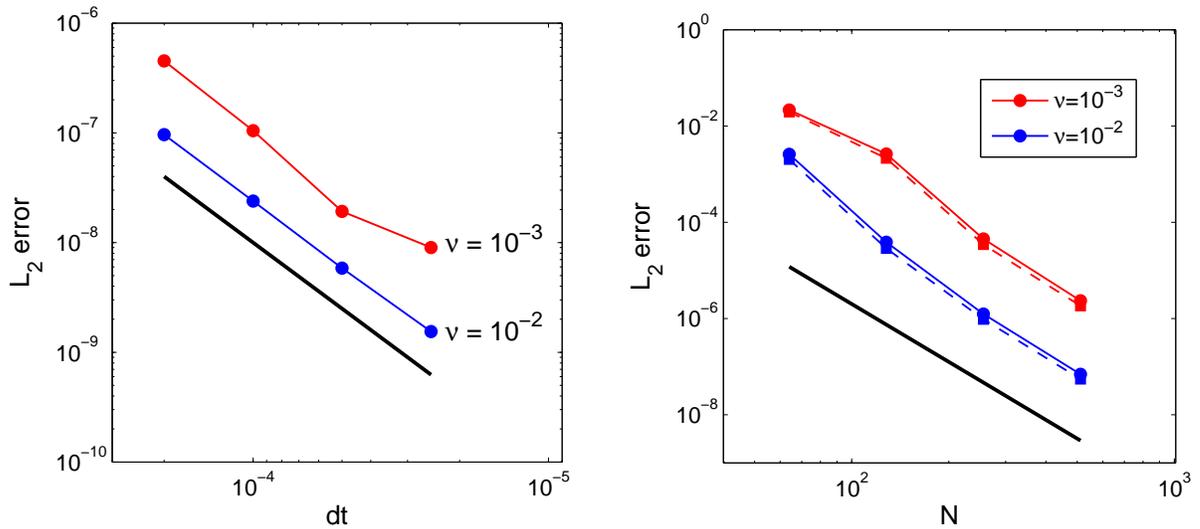


Figure 8: L_2 errors for advection-diffusion simulations. Time and spatial convergence rates are shown in the left and right frames, respectively. Sample 2nd order (left frame) and 4th order (right frame) convergence rates are shown with thick black lines for comparison purposes.

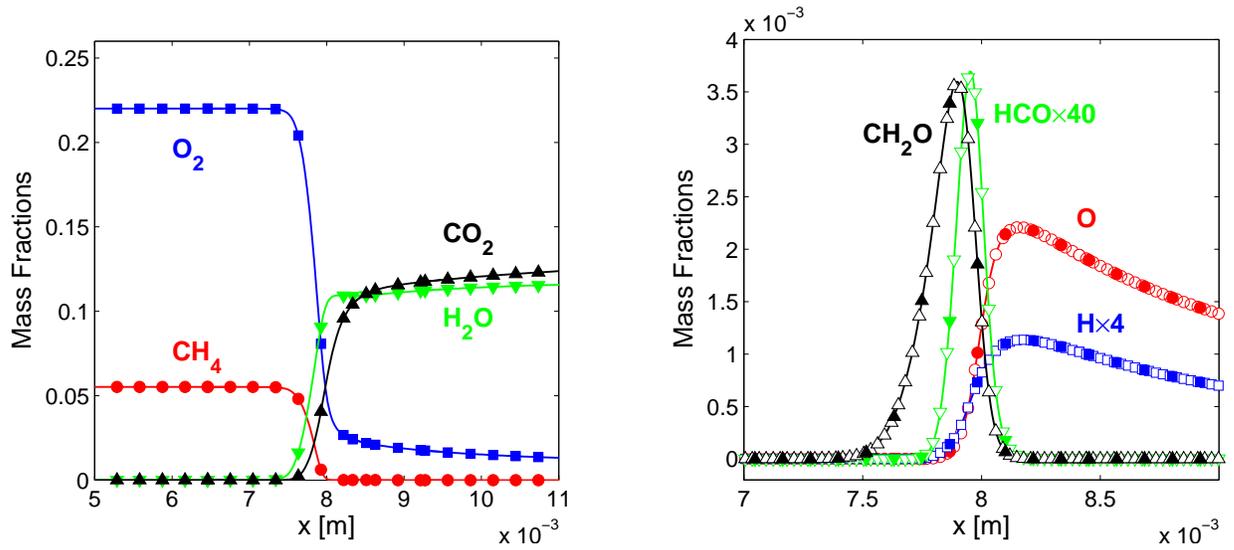


Figure 9: A sample of major species (left) and radicals (right) mass fraction profiles for a freely propagating stoichiometric CH₄-air flame. For the major species only the coarse grid solution is shown. In the right figure, the filled symbols correspond to the coarse mesh level and the open symbols to the finest mesh level.

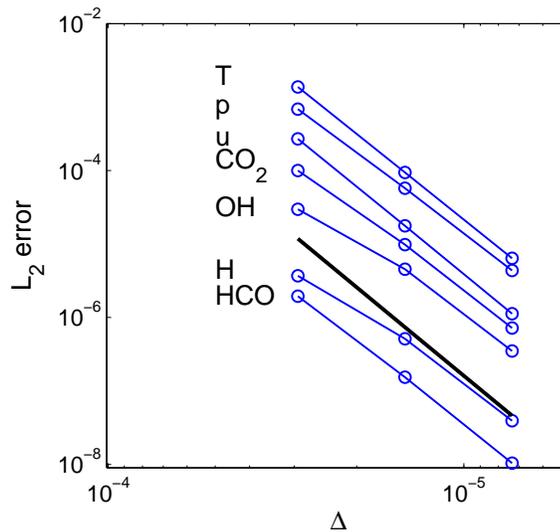


Figure 10: L_2 errors for 1D advection-diffusion-reaction simulations using 3 mesh levels. Solutions are advanced with a time step $\Delta_t = 10^{-8}$ for $t = 0.2$ ms. Errors are measured in a region $1.2T_{\min} < T < 0.8T_{\max}$, where $T_{\min} = 300$ K and $T_{\max} = 2165$ K. Sample 4th order convergence rates are shown with a thick black line for comparison purposes.

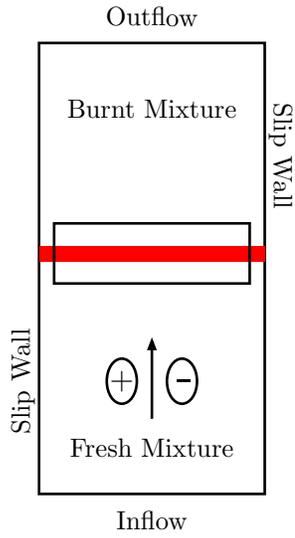


Figure 11: Schematic of the computational domain for the vortex pair-flame interaction.

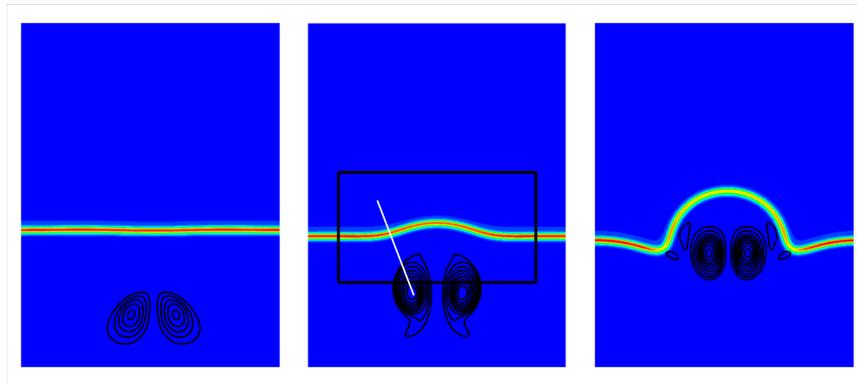


Figure 12: Vorticity contours (black lines) and heat release rates (colormap) during the interaction between a vortex pair and an initially flat premixed flame. The frames correspond to 0.05 ms, 0.3 ms, 0.5 ms from the beginning of the simulation. The simulation employed 2 mesh levels with 512×256 grid cells (or $h_x = h_y = 30\mu\text{m}$) on the coarse level.

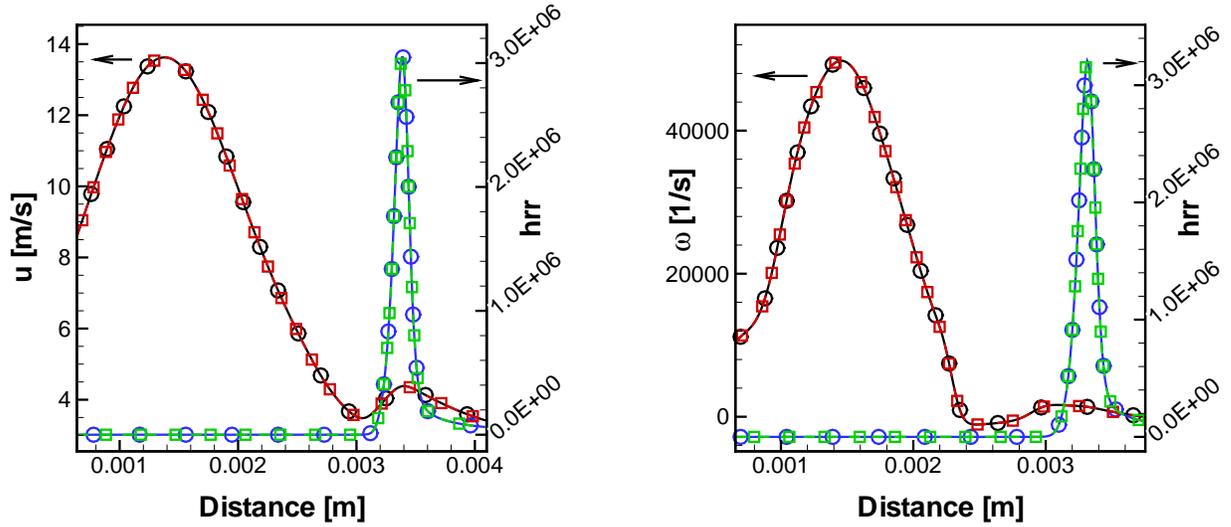


Figure 13: Left frame: x -velocity and HRR (shown with cgs units) profiles along a centerline slice. Right frame: vorticity and heat release rate profiles along a slanted slice going through the center of the left vortex and intersecting the flame at the point closest to the left vortex. These slices are taken at 0.3 ms from the beginning of the simulation. The circles indicate a solution obtained from a uniform mesh run, while the squares indicate one from an equivalent 2-level SAMR simulation.

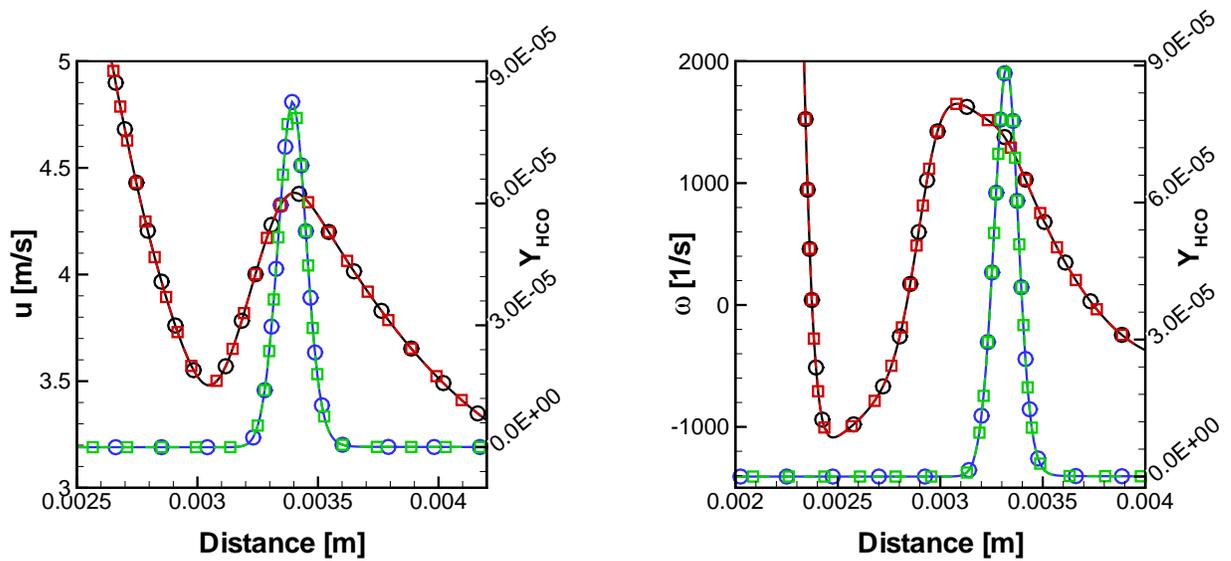


Figure 14: Left frame: x -velocity and HCO mass fraction profiles along a centerline slice through the flame. Right frame: vorticity and HCO mass fraction profiles along a slanted slice going through the center of the left vortex and intersecting the flame at the point closest to the left vortex. Results correspond to the same time as in Fig. 13. The circles indicate a solution obtained from a uniform mesh run, while the squares indicate one from an equivalent 2-level SAMR simulation.

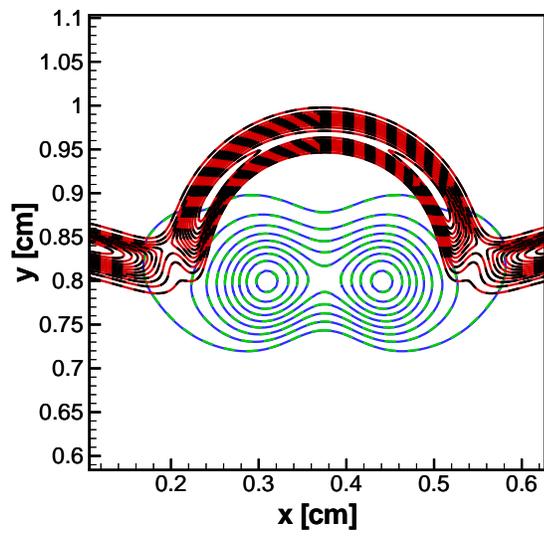


Figure 15: Pressure (blue and green contours) and HCO mass fraction (red and black contours). The blue and red contours correspond to a SAMR simulation with two levels of refinement and $h_x = 60\mu\text{m}$ on the coarse level, while the green and black contours correspond to a uniform mesh simulation with $h_x = 15\mu\text{m}$. The snapshot corresponds to 0.5 ms from the beginning of the simulation.

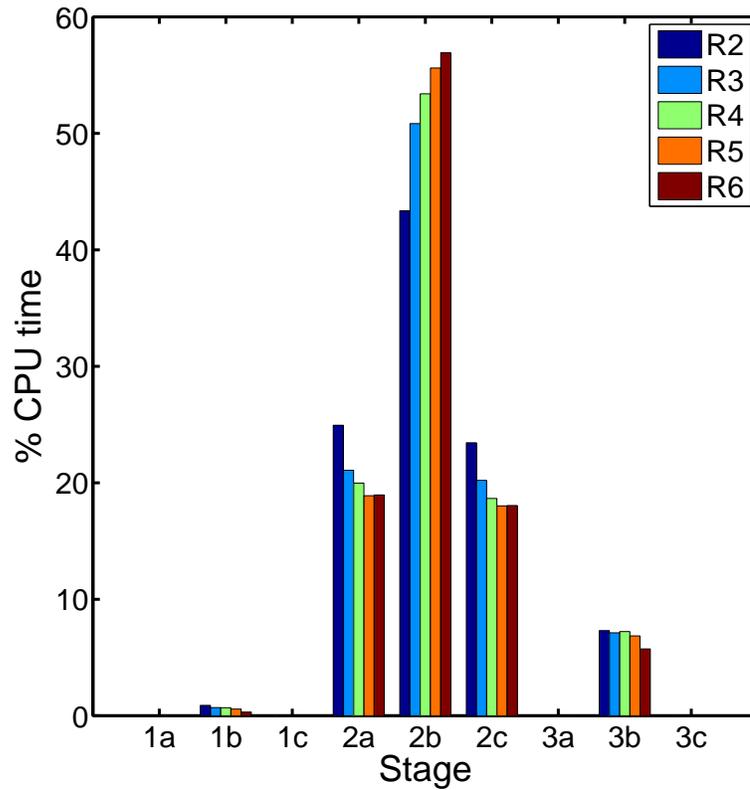


Figure 16: Computational expense of various algorithm components. The values are relative to the total CPU time for each corresponding simulation. The setup for these simulations is described in Section 4.4.

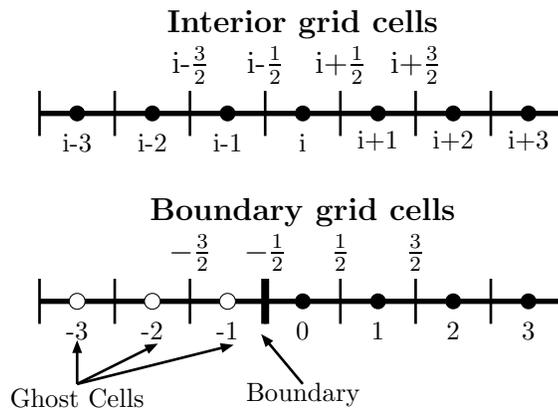


Figure 17: Cell center and face center indices inside the computational domain and near the boundary.

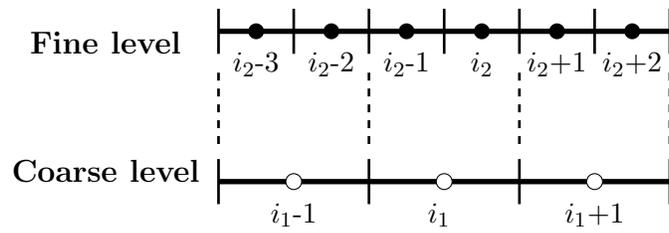


Figure 18: Correspondence between computational cells on adjacent coarse and fine grid levels.

Δt range [s]	T	ρ	u	∇p	Y_{CH_4}	Y_{O_2}	Y_{CO_2}	Y_{CH_3}	Y_{HCO}
$(50 \rightarrow 25 \rightarrow 12.5) \times 10^{-8}$	2.02	2.04	2.10	2.30	2.02	2.02	1.95	2.00	2.30
$(25 \rightarrow 12.5 \rightarrow 6.25) \times 10^{-8}$	2.06	2.07	2.10	2.50	2.05	2.05	2.02	2.03	1.67
$(12.5 \rightarrow 6.25 \rightarrow 3.125) \times 10^{-8}$	2.07	2.08	2.08	2.30	2.05	2.06	2.05	2.04	1.83

Table 1: Temporal convergence rates for $M = 16$ for 1D simulations on a uniform mesh with $\Delta_x = 7.5\mu\text{m}$. Solutions are advanced for $t = 0.4$ ms and errors are measured in a region $1.2T_{\min} < T < 0.8T_{\max}$. $T_{\min} = 300$ K is the fresh mixture temperature and $T_{\max} = 2165$ K is the temperature if the burnt gases.

Δt range [s]	T	ρ	u	∇p	Y_{CH_4}	Y_{O_2}	Y_{CO_2}	Y_{CH_3}	Y_{HCO}
$(50 \rightarrow 25 \rightarrow 12.5) \times 10^{-8}$	1.92	1.92	1.97	1.58	1.92	1.91	1.85	1.94	1.53
$(25 \rightarrow 12.5 \rightarrow 6.25) \times 10^{-8}$	2.01	2.01	2.02	1.84	2.00	1.99	1.97	2.00	1.48
$(12.5 \rightarrow 6.25 \rightarrow 3.125) \times 10^{-8}$	2.07	2.07	2.07	1.96	2.05	2.05	2.05	2.05	1.84
$(6.25 \rightarrow 3.125 \rightarrow 1.5625) \times 10^{-8}$	2.17	2.18	2.17	2.01	2.13	2.13	2.16	2.12	1.94

Table 2: Temporal convergence rates for $M = 8$ for 1D simulations using 3 mesh levels and a C1-mechanism. The grid size on the coarse mesh level is $\Delta_x = 30\mu\text{m}$. Solutions are advanced for $t = 0.4$ ms and errors are measured in a region $1.2T_{\min} < T < 0.8T_{\max}$. T_{\min} and T_{\max} are defined in Table 1.

Δx range [μm]	T	ρ	u	v	∇p_x	∇p_y	Y_{CH_4}	Y_{O_2}	Y_{CO_2}	Y_{CH_3}	Y_{HCO}
$15 \rightarrow 30 \rightarrow 60$	3.9	4.0	3.8	3.7	3.8	3.7	3.9	3.9	3.9	3.9	4.1

Table 3: Spatial convergence rates for 2D vortex-pair flame interactions using 2 level mesh and a C1-mechanism. Solutions are advanced with a time step $\Delta t = 2 \times 10^{-8}$ s and errors are measured after $t = 0.3$ ms from the beginning of the simulations.

	$R3$	$R4$	$R5$	$R6$
% CPU time $R1$	17	18	21	37

Table 4: Relative CPU times for a sequence of SAMR simulations with one refinement level ($R3$ - $R6$), compared to a uniform mesh run ($R1$).

	j	-2	-1	0	1	2	3
4 th	$c_j \times 128$	-	-7	105	35	-5	-
6 th	$c_j \times 8192$	77	-693	6930	2310	-495	63

Table 5: 4th and 6th order interior stencil coefficients for interpolation of cell-centered values from coarse to fine mesh levels.

j	0	1	2	3
$c_{0,j} \times 128$	195	-117	65	-15
$c_{1,j} \times 128$	77	77	-33	7
$c_{2,j} \times 128$	15	135	-27	5

Table 6: 3rd order boundary stencil coefficients for interpolation of cell-centered values from coarse to fine mesh levels.

j	0	1	2	3	4	5
$c_{0,j} \times 8192$	13923	-13923	15470	-10710	4095	-663
$c_{1,j} \times 8192$	4389	7315	-6270	3990	-1463	231
$c_{2,j} \times 8192$	663	9945	-3978	2210	-765	117
$c_{3,j} \times 8192$	-231	5775	3850	-1650	525	-77
$c_{4,j} \times 8192$	-117	1365	8190	-1638	455	-63

Table 7: 5th order boundary stencil coefficients for interpolation of cell-centered values from coarse to fine mesh levels.